RAZzies

Maandblad van de Radio Amateurs Zoetermeer



Augustus 2018

Colofon

RAZzies is een uitgave van de Radio Amateurs Zoetermeer. Bijeenkomsten van de Radio Amateurs Zoetermeer vinden plaats op elke tweede en vierde woensdag van de maanden september - juni om 20:00 uur in het clubhuis van de Midgetgolfclub Zoetermeer in het Vernède sportpark in Zoetermeer.

Website:

http://www.pi4raz.nl

Redactie:

Frank Waarsenburg
PA3CNO
pa3cno@pi4raz.nl

Informatie:

info@pi4raz.nl

Kopij en op- of aanmerkingen kunnen verstuurd worden naar razzies@pi4raz.nl

Nieuwsbrief:

http://pi4raz.nl/maillist/subscribe.php

Van de redactie

Het is eigenlijk geen weer om in de shack te zitten. Veel te warm. Maar ja, soms kruipt het bloed waar het niet gaan kan. Na het op het nippertje nog halen van het bronzen award voor het werken van stations die gerelateerd waren aan het WK voetbal in Rusland, heb ik me voornamelijk gestort op het werken met QRP vermogens. Daartoe is de goeie ouwe K1 weer van stal gehaald, en de resultaten zijn nog best goed, de slechte condities in aanmerking nemende. Helaas kan ik dat niet zeggen van de Spoetnik transciever, maar ja, 700mW op 15m in een Inverted-V is nou niet direct overtuigend te noemen. Zefls het Reverse Beacon Netwerk vindt het de moeite van het vermelden niet waard. Daarvoor moeten er toch echt wat betere condities zijn. Maar ook met mijn Classie wil het nog niet lukken. Dat heeft er meer mee te maken dat ik voornamelijk in het weekend tijd heb om daarmee te spelen, en aangezien weekends het domein zijn van de contesters, word je gewoon van de band gedrukt met je 1W signaaltje. Dat bewaar ik wel tot het weer wat vroeger donker wordt en de wat zwakkere signaaltjes weer wat verder dragen. De rest van de tijd wordt gevuld met het in elkaar zetten van een op Arduino gebaseerde automatische antennetuner. De spoelensets waren al een tijdje gereed, maar de stuurelektronica nog niet. Dat schiet nu lekker op. Dan nog de Directional Coupler in elkaar zetten en kijken of het werkt...

LoRa

Robert de Kok, PA2RDK

De achtergrond

Recentelijk werd ik geconfronteerd met het bestaan van het fenomeen LoRa. LoRa staat voor Long Range en is een digitale radiomode, bedoeld als de eerste laag (fysieke laag) in het OSI model en bedoeld om met weinig vermogen van zo'n 25 mW, over grote afstanden, tot wel 10km, digitale berichten te versturen.

Primair is LoRa bedoeld om met minimaal energiegebruik, daar waar niet direct internet voorhanden is, meestal batterij gevoede sensoren op afstand uit te lezen en waarbij de batterij bij voorkeur minimaal een jaar mee kan gaan.

Nuttige praktijkvoorbeelden voor dit soort sensoren zijn er voldoende, denk aan de bewaking van de locatie en conditie van een kudde koeien, een weerstation op een berg, een GPS tracker, de besturing van lantaarnpalen, de uitlezing van energie- en verbruiksmeters, de beveiliging van gebouwen en objecten etc.

Als goed radioamateur intrigeerde de specificaties mij en was ik uiteraard sceptisch maar zag ik allerlei leuke toepassingen, dus dat was reden genoeg om mij erin te gaan verdiepen.

Er zijn vele radiomodes voor het

versturen van digitale berichten en het streamen van audio en video, zoals Bluetooth en het simpele broertje BLE, Wifi, GSM etc. Al deze modes zijn bedoeld voor de korte afstand en het verplaatsen van een breedbandig signaal, zoals audio en video. LoRa is dit niet, het is bedoeld voor het verplaatsen van korte berichten en maakt daarbij gebruik van een smalbandige modulatie.

Als radioamateur weten we allemaal dat door het gebruik van een smalbandig signaal de reikwijdte veel groter is dan van een breedbandig signaal, FM heeft nu eenmaal een veel beperkter bereik dan SSB of liever nog CW. Maar in een smalbandig signaal past ook veel minder informatie dan in een breedbandig signaal, daarom is LoRa bedoeld voor korte berichten.

Het smalbandige van het LoRa signaal is niet de enige reden waarom de reikwijdte zo groot is. Er wordt ook gebruik gemaakt van spreadspectrum technologie. Bij spreadspectrum wordt een smalbandig signaal uitgezonden in een breedbandig kanaal. Een smalbandige LoRa signaal wordt hier als het ware in verspreid over een breedbandig kanaal. Dit gebeurt door de frequentie van het signaal continu te wijzigen binnen de bandbreedte van het kanaal. Er zijn meerdere spreadspectrum technieken, een ervan werkt als volgt: Door de snelheid waarmee de frequentie wordt gewijzigd veel hoger te maken dan de minimale lengte van een data bit, wordt het data bit verspreid over verschillende frequenties en dus eigenlijk meerdere malen uitgezonden. Dit verkleint de kans op storingen en dataverminking significant en draagt dus bij aan een meer betrouwbare dataoverdracht waarbij het signaal zelfs gedecodeerd kan worden als het onder het ruisniveau zit. Zie ook: https://en.wikipedia.org/wiki/Spread_spectrum

De spreadspectrum technologie heeft nog een voordeel: het signaal is slecht te herkennen in de ruis, door het verspringen van de frequentie klinkt het zelf naar ruis en is mede daarom moeilijk af te luisteren.

LoRa is door het Franse bedrijf Semtech gepatenteerd en in 2012 op de markt gebracht. Omdat het protocol gepatenteerd is, bestaat er weinig informatie over hoe het in detail werkt en zijn de chips relatief duur. Een interessant inhoudelijk artikel over hoe LoRa in werkt vind je hier: https://pubs.gnuradio.org/index.php/grcon/article/download/8/7

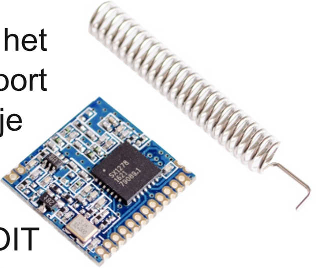
Bij Ali ben je voor een node vanaf $4,- kwijt, maar hierover later meer.

In Europa wordt LoRa toegepast in de 868MHz band, in onder andere Azië in de 433MHz en in de US op 915MHz. Let hierop als je bij Ali gaat bestellen, ik heb al de nodige dingetjes voor LoRa op 433MHz gekocht, gewoon omdat ik te ongeduldig was. Voor de eerste experimenten is het geen probleem en het mag ook gewoon in Nederland, maar verderop zal duidelijk worden dat je er verder weinig mee kunt.

De eerste experimenten

Nog niet volledig gewapend met bovenstaande kennis, maar wel erg nieuwsgierig, ben ik gaan bestellen bij Ali. In eerste instantie 2 radio modules voor de 433MHz band. Zoals al eerder gemeld, het patent van de chips ligt bij Semtech, dus alle modules hebben een Semtech chip aan boord, meestal een SX127x. De modules zijn voor zowel RX als TX geschikt.

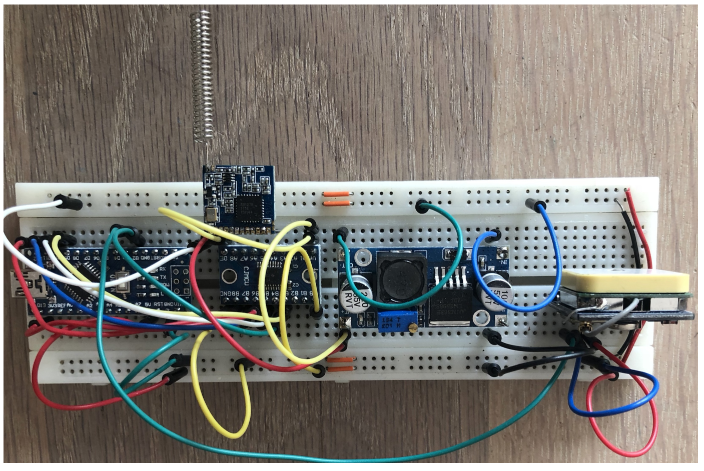
Zoals op de foto te zien een bescheiden formaat van nog geen 2x2 cm.



Als zendamateur zijn we het wel gewend, maar bij dit soort kleine modules verwacht je het niet, in alle documentatie wordt gewaarschuwd de modules NOOIT te gebruiken zonder antenne.

Ik heb het zo vaak gelezen en ben het dus maar serieus gaan nemen. De modules hebben een SPI-interface en zouden dus direct op een Arduino moeten kunnen worden aangesloten, ware het niet dat mijn favoriete Arduino, een Nano met Atmega328 chip, een voedingsspanning van 5 volt wil zien en de LoRa module het met maximaal 3,3 volt doet.

Behalve de SPI-interface (MOSI, MISO, en GND) heeft de module nog een aantal aansluitingen, waaronder Reset, NSS (chip select) en 3 I/O poorten. Kortom, een levelconverter met voldoende I/O was noodzakelijk om de module aan te sluiten op de Arduino. Nu heeft een Arduino een 3,3V uitgang, maar deze kan maar beperkt stroom leveren en volgens de specificaties van de LoRa module kan er met pieken wel stroom gevraagd worden. Dus het breadboard maar voorzien van een dc/dc-converter om van de 5V 3.3V te maken. Op de foto zie je ook nog een GPS-module zitten, omdat ik een GPS tracker als startproject wel een leuke uitdaging vond.



Uiteraard heb ik op deze manier 2 breadboards gevuld, want er is zowel een RX als TX-station noodzakelijk om te kunnen testen. Wellicht vraag je je af, waarom staat er hier geen schema, maar daar heb ik een goede reden voor. Natuurlijk kan je het gewoon krijgen, maar de route zoals ik hem heb bewandeld is niet de handigste en ook niet de goedkoopste. Hierover straks meer.

Eerst wat software, op het internet is een library voor de Arduino beschikbaar voor het aansturen van de LoRa chips. Als je deze library installeert, zal je bij de voorbeelden een LoRaSender en LoRaReceiver vinden. Zoals te zien in de tekstblokken hier rechts, het stelt helemaal niets voor en het werkte direct zonder enig probleem.

De LoRa zender zond elke 5 seconden een bericht en de LoRa receiver ontving deze en zette het ontvangen bericht op de seriële poort.

LoRaSender

```
#include <SPI.h>
#include <LoRa.h>

int counter = 0;

void setup() {
  Serial.begin(9600);
  while (!Serial);

  Serial.println("LoRa Sender");

  if (!LoRa.begin(868E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
}

void loop() {
  Serial.print("Sending packet: ");
  Serial.println(counter);

  // send packet
  LoRa.beginPacket();
  LoRa.print("hello ");
  LoRa.print(counter);
  LoRa.endPacket();

  counter++;

  delay(5000);
}
```

Code voor de zender

LoRaReceiver

```
#include <SPI.h>
#include <LoRa.h>

void setup() {
  Serial.begin(9600);
  while (!Serial);

  Serial.println("LoRa Receiver");

  if (!LoRa.begin(868E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
}

void loop() {
  // try to parse packet
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    // received a packet
    Serial.print("Received packet '");

    // read packet
    while (LoRa.available()) {
      Serial.print((char)LoRa.read());
    }

    // print RSSI of packet
    Serial.print("' with RSSI ");
    Serial.print(LoRa.packetRssi());
    Serial.print("' and SNR ");
    Serial.println(LoRa.packetSnr());
  }
}
```

Code voor de ontvanger

Dit is trouwens in vergelijking met andere, eerder genoemde, communicatieprotocollen een belangrijk aspect van LoRa, het is geschikt voor peer-to-peer communicatie zonder tussenkomst van een server. Saillant detail, ik was hiermee in Liechtenstein aan het spelen en uiteraard waren we erg nieuwsgierig naar het bereik van de modules. Piet verklaarde zich bereid een wandeling te gaan maken, dus de TX-module aan een accupack en Piet op stap. De resultaten vielen niet tegen, 10 km was het niet, maar een paar km lukte wel zonder degelijke antennes en met de ontvanger binnen.

Zo eenvoudig kan de LoRa wereld niet zijn, dus heb ik mij verdiept in de mogelijkheden van de library. Vaak wordt zo’n library beschouwd als een black box, maar niets is minder waar. Het is ook gewoon C++ code en met een beetje fantasie is het vaak prima te lezen.

De reden dat bovenstaande code werkt is dat beide chips dezelfde standaard instelling hebben en elkaar dus begrijpen. Maar een LoRa signaal heeft een aantal parameters welke overeen moeten komen:

- De frequentie. Duh…
- De bandbreedte. Hoe smaller de bandbreedte hoe trager de overdracht maar hoe verder de overdracht van het signaal. De bandbreedte varieert in stappen van 7,8 tot 500 kb, waarbij 125 kb de standaard is.
- De spreadingfactor, dit zijn het aantal frequenties waarmee de spreadspectrum functionaliteit werkt. Deze heeft een waarde van tussen de 7 en 12.
- De Codingrate of forward error correction, een waarde tussen de 5 en 8. LoRa ondersteunt 4/5, 4/6 ,4/7 en 4/8. Standaard wordt altijd 4/5 gebruikt.
- Het Syncword, hierbij wordt 0x12 voor privé en 0x34 voor publieke (LoRaWAN) toepassingen gebruikt. Standaard staat de chip op 0x12, dus goed voor de bovenstaande code. Over LoRaWAN later meer.

Als een van deze parameters niet overeenkomt, gaat communiceren niet lukken. Dus laten we de

```
LoRa.setSignalBandwidth(125E3);
LoRa.setSpreadingFactor(9);
LoRa.setCodingRate4(5);
LoRa.setSyncWord(0x12);
LoRa.enableCrc();
```

setup maar een beetje uitbreiden:
Een LoRa bericht kan voorzien worden van een CRC check, zodat aan de ontvangende kant gecontroleerd kan worden of het bericht correct is ontvangen.

Een voorbeeld dat zeker ook de moeite waard is om te bekijken is de LoRaReceiverCallback. Hierbij wordt gebruik gemaakt van een programmeermodel welke je regelmatig terugziet bij communicatie libraries of andere libraries waarbij vanaf buiten wordt getriggerd. In de setup wordt de LoRa module geïnitialiseerd, zoals we ook in de andere voorbeelden hebben gezien, echter aanvullend wordt een ‘onReceive’ commando gegeven. Hierbij wordt er gebruik gemaakt van de interrupt faciliteiten van de processor. Zodra de LoRa ontvanger een bericht ontvangt wordt er een interrupt afgeschoten naar de processor. Met het ‘onReceive’ commando hebben we de processor uitgelegd dat als er zo’n interrupt geschiedt de processor moet stoppen met wat die aan het doen is en de subroutine ‘onReceive’ moet uitvoeren. Daarom is de loop ook leeg, er wordt niets geloopt, maar zodra een bericht binnenkomt wordt dit verwerkt. Deze methodiek stelt je in staat een programma te schrijven dat zijn werk doet en alleen naar de LoRa module gaat luisteren als er daadwerkelijk data klaar staat.

Een processor heeft meerdere interrupt mogelijkheden, waaronder tijd gestuurde interrupts. Deze worden bijvoorbeeld gebruikt voor klok en tellerfuncties, maar ook interrupts als gevolg van het veranderen van de status van een input lijn.

Het gebruik van de ‘onReceive’ callback activeert het interrupt mechanisme in de processor. Dit kan consequenties hebben voor de rest van het programma. Het gebruik van

onReceive, Callback voorbeeld

```
#include <SPI.h>
#include <LoRa.h>

void setup() {
  Serial.begin(9600);
  while (!Serial);

  Serial.println("LoRa Receiver Callback");

  if (!LoRa.begin(868E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }

  // register the receive callback
  LoRa.onReceive(onReceive);

  // put the radio into receive mode
  LoRa.receive();
}

void loop() {
  // do something or nothing
}

void onReceive(int packetSize) {
  // received a packet
  Serial.print("Received packet '");

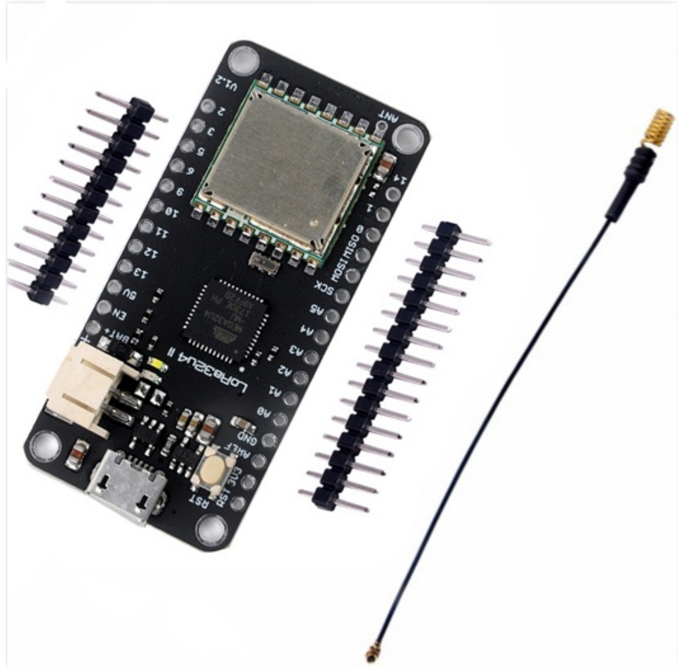
  // read packet
  for (int i = 0; i < packetSize; i++) {
    Serial.print((char)LoRa.read());
  }

  // print RSSI of packet
  Serial.print("' with RSSI ");
  Serial.println(LoRa.packetRssi());
}
```

interrupts, de juiste instelling van de interrupt masks, het zorgen dat een interrupt routine de state van het lopende programma niet verknalt en interrupts elkaar niet in de weg zitten, wordt niet ervaren als het meest eenvoudige deel van het programmeren van micro controllers.

De Arduino omgeving heeft het gebruik en de afhandeling van interrupts wel sterk vereenvoudigd, maar beschermt slecht tegen het verkeerd gebruik van interrupts. Maar het is het waard je in de materie te verdiepen, middels interrupts lukken veel dingen wel die in een gewone programma loop niet lukken.

Een mooi voorbeeld is de afhandeling van drukknoppen of een rotary encoder. Als de state van een drukknop in de loop wordt uitgelezen en de hele loop duurt één seconde, dan kan het zomaar zo zijn dat de knop een seconde moet worden ingedrukt voor het programma reageert. Door de drukknop aan een I/O lijn met interrupt functie te binden (D2 en D3 op een 328) wordt bij het indrukken van de knop een interrupt gegenereerd en de functie in de interrupt subroutine uitgevoerd. Natuurlijk gaat dit niet vanzelf en dien je de functionaliteit te programmeren, maar dit valt buiten de scope van dit artikel.



Voor rond de €5,- koop je een LoRa module voor 868MHz met een SX1276 chip er op. Je hebt dan nog een levelconverter, een Arduino en een DC/DC converter nodig om het aan het werk te krijgen. Ik vond een eenvoudigere oplossing, namelijk deze:
http://bit.ly/2lw6WGn
Dit is een Arduino met een 32u4 processor, een LoRa module en een LiPo laadcircuit. Het grootste verschil tussen een 328 en een 32u4 is dat de 32u4 USB aan boord heeft en de 328 dat met een externe chip moet regelen. Beiden hebben 32k flash aan boord, maar in de praktijk is in de 32U4 een beetje minder geheugen beschikbaar. Ik vermoed door een grotere bootloader, maar ik heb mij er nooit in verdiept. In de Arduino ontwikkelomgeving kies je voor een ander board, de Leonardo in plaats van de Nano. Vrijwel alle sketches voor de Nano werken ook probleemloos in de Leonardo. Dus dat scheelt een hoop gepriegel. De module kost rond de €13,-. Dat scheelt niet heel veel met de alle losse onderdelen rondom een losse LoRa module. De LoRa module zit vast aan de

processor en dus heb je geen invloed op de bedrading. Dit moet in de code worden geregeld. In plaats van de LoRa library, wordt er gebruik gemaakt van de LoRa32u4 Library. Volgens mij is het verschil nihil, maar ik ben er nog niet aan toe gekomen om de 32u4 te testen met de gewone LoRa library.

Een interessant verschil in de initialisatie van de LoRa module is de regel ‘LoRa.begin(868E6)’ Dit is de wetenschappelijke notatie en er wordt bedoeld:
$868e6 = 868*10^6 = 868000000$ = 868MHz.
In het voorbeeld hiernaast staat ‘LoRa.begin(868300000,true)’ Hierbij wordt er een exacte frequentie gezet, waarbij in het eerste voorbeeld alleen de band wordt gezet en de default frequentie van de chip in die band wordt geselecteerd.

De code voor de LoRa TX module bestaat natuurlijk ook. Alle voorbeelden en de gebruikte libraries zijn [HIER] te downloaden.

Voor allerlei toepassingen waarbij er gecommuniceerd dient te worden met en tussen devices, zoals bijvoorbeeld bij weerstations en domotica toepassingen, zijn bovenstaande voorbeelden voldoende basis om mee verder te kunnen bouwen. Zoals eerder genoemd, LoRa modules kunnen zowel zenden als ontvangen, dus het is mogelijk om betrouwbare en controleerbare peer-to-peer communicatie op te zetten. Realiseer je wel dat standaard LoRa pakketten niet zijn beveiligd en kunnen worden meegelezen door iedereen binnen het bereik

LoRa RX module voor de 32U4

```
#include <SPI.h>
#include <LoRa32u4.h>

//LoR32u4II 868MHz or 915MHz (black board)
#define SCK     15
#define MISO    14
#define MOSI    16
#define SS      8
#define RST     4
#define DIO     7
#define BAND    868E6  // 915E6
#define PABOOST true

void setup() {
        delay(5000);
        Serial.begin(9600);
        //while (!Serial);
        Serial.println("LoRa Receiver");
        LoRa.setPins(SS,RST,DIO);
        if (!LoRa.begin(868300000,PABOOST)) {
                Serial.println("Starting LoRa failed!");
                while (1);
        }
        LoRa.setSignalBandwidth(125E3);
        LoRa.setSpreadingFactor(9);
        LoRa.setCodingRate4(5);
        LoRa.setSyncWord(0x34);
        LoRa.enableCrc();
}

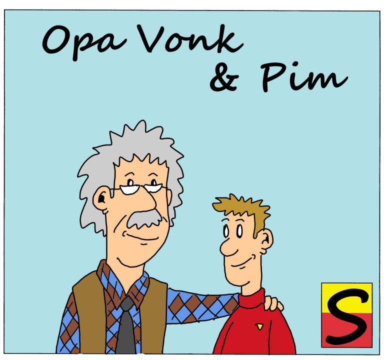
void loop() {
        // try to parse packet
        int packetSize = LoRa.parsePacket();
        if (packetSize) {
                // received a packet
                Serial.print("Received packet '");

                // read packet
```

van de LoRa zender, en dit kan wel 10km zijn! Als je de communicatie dus wilt beveiligen, dien je dit zelf in het programma te regelen.

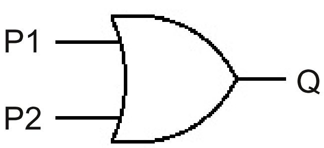
Een LoRa module met bijvoorbeeld een temperatuur sensor of een GPS ontvanger noemen we een LoRa Node. Er bestaan commerciële nodes om van alles te kunnen meten en registreren, maar met bovenstaande kennis kun je ook zelf iets leuks bouwen!

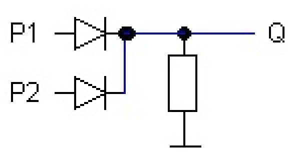
Maar de bedoeling van LoRa is veel groter, daarom in de volgende RAZzies verder over LoRaWAN…



Pim keek met grote ogen naar een schema waar zijn Opa Vonk mee bezig was. "Wat zijn dat allemaal?", vroeg hij, wijzend op een aantal vreemde vormen op het schema waar diverse draden naartoe liepen. "De meeste van die schema's van u kan ik nu wel lezen, maar dit heb ik nog nooit gezien". Opa keek Pim over zijn leesbril aan. "Dat zijn NAND en OR poorten", antwoordde hij. "Ik bouw meestal analoog, maar met het voortschrijden van de techniek kom je niet meer onder digitale technieken uit. De basisbouwstenen van de digitale techniek zijn de poorten: in het Nederlands EN, NEN, OF en NOF, maar we zullen de Engelse termen

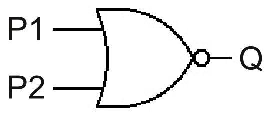
gebruiken omdat je die het vaakst tegenkomt. En dat is AND, NAND, OR en NOR. Ooit van gehoord?" vroeg Opa. "Gehoord wel", zei Pim. "Maar nooit iets mee gedaan. In een radio kom je dat zogauw niet tegen". "Fout", zei Opa. "Zeker in een SDR zit een hoop digitale logica. Denk maar eens aan de simpele SDR ontvanger voor 40m uit de RAZzies van maart 2017. Daar zat een heleboel van dat soort logica in. Maar weet je hoe het werkt?" vroeg Opa. Pim aarzelde, maar schudde toen zijn hoofd. "In grote lijnen. Denk ik." zei Pim. "Dat dacht ik wel. Ik zal je de basisbeginselen bijbrengen. En ik gebruik daarvoor de Amerikaanse symbolen, omdat je die het meeste ziet in schema's. Bij poorten in de digitale techniek heb je minimaal 1 ingang en 1 uitgang. Meestal zijn er meerdere ingangen. De enige poorten die 1 ingang en 1 uitgang hebben, zijn inverters. In het Nederlands: omkeerders. Als je daar een logische 1 op zet, is de uitgang 0, en andersom. Wat die logische 1 is, hangt af van de gebruikte techniek. Dat kan 5V zijn, maar ook 12V. Het is maar net wat je een 1 noemt. Zijn er meerdere ingangen aanwezig bij een digitale poort, dan is de uitgang afhankelijk van de waarde van de ingangen. Die afhankelijkheid wordt dan weergegeven in een functietabel of waarheids-tabel. Die beschrijft wat de uitgang doet als resultaat van de ingangen. Laten we eerst eens kijken naar een OR poort met twee ingangen. In zijn eenvoudigste vorm zijn dat twee dioden waarvan de kathodes aan elkaar liggen.

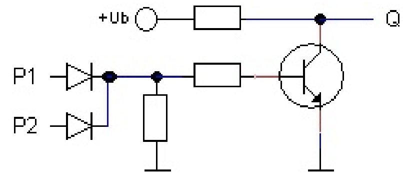




Ingangen		Uitgang
P_1	P_2	Q
0	0	0
0	1	1
1	0	1
1	1	1

Hoe het werkt, is makkelijk te zien. Als op één van de ingangen P spanning wordt gezet (een 1), zal de uitgang Q ook een spanning voeren (een 1). In de waarheidstabel is dat goed te zien; die geeft de waarde van uitgang Q voor alle mogelijke combinaties van P1 en P2, en dat zijn er vier. Als óf P1 een 1 is, óf P2 een 1 is, dan is de uitgang ook een 1. Als P1 en P2 beiden 1 zijn, dan is Q uiteraard ook al 1, want er hoeft maar één ingang 1 te zijn om de uitgang ook 1 te laten zijn. Daarom heet het dus een OR (OF) poort. In het vereenvoudigde schemaatje met de twee dioden is duidelijk te zien hoe het werkt. Maar soms wil je net de omgekeerde functie: een Niet-OR ofwel een NOR poort. Die ziet er als volgt uit:





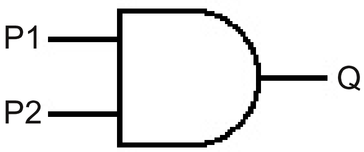
Het subtiele verschil tussen het symbool voor een OR en een NOR poort is het cirkeltje in de uitgang. Dat cirkeltje betekent dat het signaal geïnverteerd wordt, ofwel omgekeerd. Een 0 wordt een 1 en andersom. Kijk maar eens naar de waarheidstabel van de NOR poort:

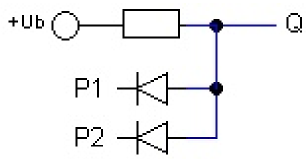
Ingangen		Uitgang
P_1	P_2	Q
0	0	1
0	1	0
1	0	0
1	1	0

De ingangswaarden van P1 en P2 zijn precies hetzelfde, maar de bijbehorende waarde voor uitgang Q is tegengesteld. Waar bij de OR poort een 0 stond, staat bij de NOR poort een 1 en omgekeerd. In de voorbeeldopbouw met discre-te componenten zie je dat er ten opzichte van de OR poort alleen een transistor is toegevoegd die voor het inverteren zorgt. In werkelijkheid zit de poort complexer in elkaar; deze schemaatjes zijn slechts ter illustratie toegevoegd.

Een OR of NOR poort kan ook meer dan 2 ingangen hebben. Daar geldt dan weer hetzelfde voor: zodra er één ingang een 1 is, is de uitgang dat ook (OR), of de uitgang wordt 0 (NOR). In de voorbeeldschema's betekent dat een diode toevoegen voor elke extra ingang.

Behalve de OR schakeling is er ook een AND schakeling. Was bij de OR de uitgang een 1 als OF P1, OF P2 (OF allebei) een 1 was, bij de AND moet EN P1, EN P2 een logische 1 zijn om een 1 aan de uitgang te krijgen. Zie weer het symbool met bijbehorende schemaatje. Alweer: het schema geeft niet het inwendige van de poort weer, maar is slechts illustratief voor de werking van een AND schakeling.





Aan het schema is makkelijk te zien dat als één van de dioden naar 0 getrokken wordt, de uitgang ook 0 is. EN P1, EN P2 moeten 1 zijn om Q 1 te laten worden. Dat is te zien in de bijbehorende waarheidstabel:

Ingangen		Uitgang
P_1	P_2	Q
0	0	0
0	1	0
1	0	0
1	1	1

Om de uitgangen van poorten te kunnen vaststellen zonder waarheidstabellen te gebruiken, wordt ook wel een rekenkundig ezelsbruggetje gebruikt. Voor een OR poort kan je dan de ingangen bij elkaar optellen om te zien wat de uitgang wordt:

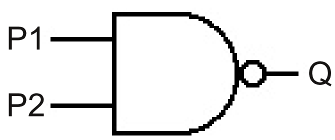
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 1

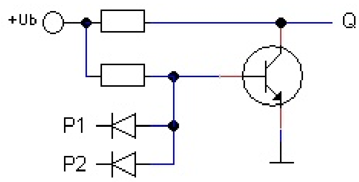
Die laatste is in de wiskunde natuurlijk niet waar: 1 en 1 is 2. Maar een logische poort kan geen 2 worden aan de uitgang, want 2 bestaat niet in het binaire stelsel. Alleen 0 of 1. Dus als het meer is dan 0, is het 1. Deze methode geldt ook voor poorten met meerder ingangen.

Voor de AND poort geldt dat je de ingangen met elkaar mag vermenigvuldigen als je wilt weten wat de uitgang wordt:

0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x 1 = 1

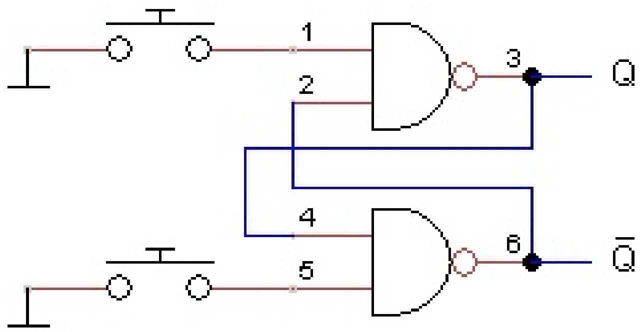
Controleer deze methode maar met de waarheidstabellen en je zult zien dat het klopt. Het inverterende broertje van de AND poort is de NAND poort:





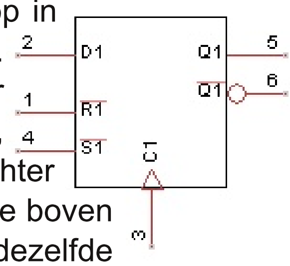
Ingangen		Uitgang
P_1	P_2	Q
0	0	1
0	1	1
1	0	1
1	1	0

Ten overvloede: het schemaatje is ter illustratie om te laten zien hoe de NAND functie werkt. In de praktijk zou dat niet betrouwbaar werken omdat je de basisspanning niet laag genoeg krijgt op deze manier. De diodes zijn immers niet ideaal, waardoor de transistor niet dicht gaat. Met deze poorten heb je de bouwstenen van de digitale techniek te pakken. Hiermee kan je allerlei schakelingen maken die een bepaalde functie hebben. Op de volgende bladzijde zijn twee NAND poorten gebruikt om een schakelaar te maken, aangestuurd door twee drukknoppen:

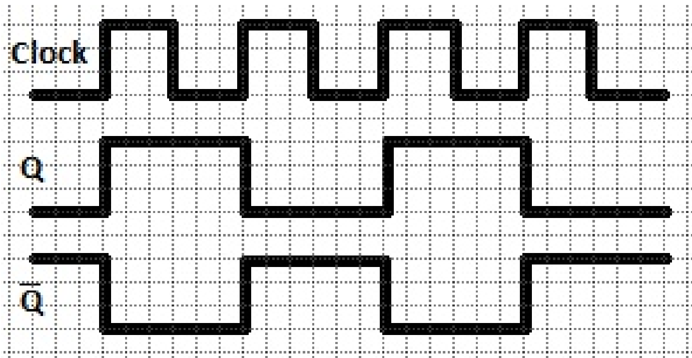


Wat er nu gebeurt is het volgende: je ziet dat de uitgang van de ene poort verbonden is met een ingang van de andere poort. Daarbij moet je nog weten dat als je een ingang nergens mee verbindt, hij als logische 1 wordt gezien. Dat is niet met alle gebruikte poorttechnieken zo, maar over het algemeen wel. Stel dat bij het inschakelen van de voedingsspanning de uitgang van de eerste poort een 0 was. Dan is één van de ingangen van de tweede poort (4) ook een 0. En volgens de waarheidstabel: als ook maar één van de ingangen 0 is, is uitgang (6) een 1. Op (2) van de eerste poort staat dus 1, en omdat (1) nergens mee verbonden is, is die ook 1. Beide ingangen van de eerste poort zijn nu 1 en daardoor is uitgang Q 0. Dit noemen we een stabiele toestand. Als we nu op de bovenste drukknop drukken, wordt (1) een 0. Ongeacht wat de waarde van (2) is, wordt uitgang Q nu een 1. Daardoor zijn van de tweede poort beide ingangen (4) en (5) nu een 1. De uitgang (6) wordt nu 0, en als je de drukknop loslaat blijft uitgang Q nu 1 omdat (2) via (6) op 0 blijft. De toestand is wederom stabiel, alleen zijn de uitgangen nu omgeklapt. Door op de onderste drukknop te drukken vindt het omgekeerde plaats: (6) wordt nu 1 en daardoor wordt Q weer een 0. Dit noem je een "flip-flop", vanwege de twee stabiele toestanden. Gelukkig hoef je de meer complexe functies niet zelf te maken: in de verkrijgbare reeks digitale IC's zijn ook complete flip-flops te koop, met verschillende functies.

Daar wil ik er eentje van noemen omdat je 'm heel vaak tegen komt: de D-flipflop. Voor ik daar op in ga, moet je weten dat er verschillende manieren zijn waarop een IC aangestuurd kan worden. In de hiervoor beschreven flip-flop is dat niveau gestuurd. Hou je de drukknop ingedrukt, dan blijft de poort waar hij aan hangt met zijn uitgang op 1, ongeacht wat de rest van de schakeling doet. Er bestaat echter ook zoiets als flank gestuurd. Wat bedoelen ze daarmee: Dat betekent dat een IC reageert op een *verandering* van het ingangssignaal, bijvoorbeeld van laag naar hoog (opgaande flank), of van hoog naar laag (neergaande flank). Is het signaal eenmaal van niveau veranderd, dan maakt dat voor het functioneren van het IC niet meer uit. Alleen de verandering zorgt voor een actie. De D-flip-flop type 7474 werkt op die manier. Je ziet deze flip-flop in het tekeningetje hier rechts. In een 7474 IC zitten er twee, en dit is de eerste, vandaar het cijfer 1 achter de aansluitingen. Het streepje boven de aansluiting heeft dezelfde betekenis als het rondje aan een uitgang: dat de werking omgekeerd is. Dat zie je bij R1 en S1: zet je er een 0 op dan wordt deze omgekeerd, dus een 1 en is R actief. De twee ingangen R (van Reset) en S (van Set) hebben dezelfde functie als de twee drukknoppen in het schema links boven. Maak je S een 0 (die dan omgekeerd wordt dus een intern een 1), dan wordt uitgang Q hoog en Q-niet (met het streepje) laag. En maak je R een 0, dan wordt Q laag en Q-niet hoog. Tot zover niets nieuws onder de zon. Het bijzondere zit 'm in de aansluitingen C en D. C staat voor Clock en D staat voor Data. Zijn S en R hoog, dus niet actief, dan is de flip-flop in rust. Als je nu de Clock van laag naar hoog laat gaan, dan neemt Q de waarde aan die D op dat moment heeft. Is D dus hoog op het moment dat de Clock hoog wordt, dan wordt Q ook hoog. En was D laag op het moment dat de Clock van laag naar hoog ging, dan wordt - of blijft - Q laag. Als je na die Clock verandering D verandert, dan maakt dat niets meer uit. Ook als je de Clock naarna laag maakt, verandert er niets, ook al is D van waarde veranderd. Q bewaart dus het niveau van D op het moment dat de Clock hoog wordt. Handig niet?" zei Opa. "Ik zie nog niet wat er handig aan is", zei Pim. "Nou, let op. Ik ga Q-niet met de D-ingang verbinden. Als Q laag is, is Q-niet dus hoog en



dientengevolge D ook. Gaat de Clock nu van laag naar hoog, dan wordt Q hoog. D was immers hoog op het moment dat de Clock hoog werd. Maar nu wordt Q-niet laag, en D dus ook. Dat verandert niets meer aan Q. Tot het moment dat de Clock weer hoog wordt. Wat je nu ziet, is dat voor elke twee keer dat de Clock hoog wordt, Q één keer hoog wordt. Je hebt dan een tweedeler gemaakt! Kijk maar:



Er zijn vele digitale IC's die allerlei functies hebben waarmee je een schakeling kunt ontwerpen. Vroeger, voor de tijd van het internet, hadden we databoeken waar je de selectie van voor jou handige IC's kon selecteren. Maar nu kan je dat met Google wel oplossen. Wat ik je nog wel bij wil brengen, zijn de verschillende technieken waarmee poorten gerealiseerd worden. Ik noemde de 7400 serie, waar de 7474 deel van uitmaakt, en dat zijn TTL IC's, waarbij TTL staat voor Transistor-Transistor logica. Deze techniek is het langst in gebruik en voldoen voor de meeste toepassingen. Een wat minder bekende techniek is ECL: Emitter Coupled Logic. Deze techniek wordt voornamelijk toegepast in situaties waar hoge snelheden belangrijk zijn. Dan heb je nog MOS: Metal Oxide Semiconductor Logic. Dit wordt toegepast in systemen met een hoge componentendichtheid. En tenslotte, en die ken je vast wel, heb je CMOS: Complementary Metal Oxide Semiconductor Logic. Deze techniek is geschikt voor systemen met heel laag stroomverbruik en begint steeds meer de belangrijkste techniek te worden. Om het nog ingewikkelder te maken, zijn deze vier hoofdcategorieën ook nog eens onderverdeels in subcategorieën. Ik noemde als typenummer voor de D-flipflop de 7474. Maar er is ook een 74LS74. Of een 74ALS74. En die hebben allemaal weer verschillende eigenschappen. Neem bijvoorbeeld de TTL serie. Die kent de volgende families:

Type	Energieverbruik	Snelheid
LS	Laag	Traag
ALS	Laag	Middel
S	Middel	Snel
AS	Middel	Snel
L	Laag	Middel
F	Hoog	Zeer snel

De 74 serie digitale IC's is ook beschikbaar in CMOS technologie. Naast de 74 CMOS serie is er de bekende CD4000 familie, die ook diverse digitale bouwstenen kent in CMOS uitvoering. Deze CMOS familie heeft de volgende eigenschappen:

Type	Energieverbruik	Snelheid
HC/HCT	Zeer laag	Middel
AC/ACT	Laag	Snel
C	Zeer laag	Middel
LCX/LVX/LVQ	Zeer laag	Middel
VHC/VHCT	Laag	Snel
CD4000	Laag	Middel

Verder moet je er rekening mee houden dat je niet eindeloos poorten aan een uitgang van een enkele poort kunt verbinden. Er is een limiet aan het aantal ingangen dat een uitgang kan aansturen. Dat heet Fan-Out. De Fan-Out is dus het maximaal aantal poorten dat nog aangestuurd kan worden waarbij de definities van de logische 1 en 0 nog gehaald kunnen worden. Over spanningen en stromen gesproken: dat verschilt ook nogal tussen de families. Zie daarvoor de onderstaande tabel. Daarin heb ik een paar parameters voor je uitgewerkt. IIH is de stroom die een poort vraagt om HOOG getrokken te worden. En zo is IIL de stroom die nodig is om een ingang LAAG te

	IIH	IIL	IOH	IOL	VIH	VIL
TTL	40uA	1,6mA	400uA	16mA	2,0	0,8
HC	1uA	1uA	4mA	4mA	2,5	2,2
HCT	1uA	1uA	4mA	4mA	1,6	1,2
CD4000	100nA	100nA	1mA	1mA	3,5	1,5

maken. IOH is de stroom die een poort kan leveren als hij HOOG is. En IOL is de stroom die een poort kan opnemen als hij LAAG is (bijvoorbeeld voor het aansturen van een LED). VIN is de ingangsspanning die je minimaal moet aanbieden om de poort een logische 1 te laten zien, en VIL is de maximale spanning die je kunt aanbieden om de poort een logische 0 te laten zien. En die tabel biedt je een aantal inzichten. Om te beginnen zie je dat bij een TTL poort het verschil tussen IIH en IOH enerzijds en IIL en IOL anderzijds een factor 10 is. Met andere woorden: de Fan-Out is 10. Een TTL uitgang kan dus niet meer dan 10 TTL ingangen aansturen. Dat verschil is bij de HC en HCT serie een factor 4000. Betekent het dan dat je met een enkele HC(T) uitgang 4000 ingangen aan kunt sturen? Qua stroom wel. Maar het probleem waar je veel eerder tegenaan loopt, is de ingangscapaciteit van een poort. Die is typisch ongeveer 7,5pF en als je dat maal 4000 doet, kom je op 30nF en dat gaat een beperkende factor worden. Wat je ook ziet aan de tabel, is dat een CD4000 poort niet in staat is om een TTL poort aan te sturen! Want de IIL van een TTL poort is 1,6mA, en de IOL van een CD4000 is maar 1mA en dat is niet genoeg. Ook zie je een verschil in spanningen waarbij de ingang van een poort reageert: een HCT 'ziet' al een logische 1 bij 1,6V, waarbij een CD4000 pas een logische 1 ziet bij 3,5V. Je kunt poorten van verschillende families dus niet zomaar door elkaar gebruiken. Zo kan je met een TTL IC dus best een LED aansturen die aan de voedingsspanning hangt en naar massa getrokken kan worden, want dan kan het IC 16mA leveren. Maar een LED voeden vanuit een TTL poort (dus met de LED aan massa) gaat weer niet, omdat een TTL poort maar 400uA (en dat is 0,4mA) kan leveren. Daarentegen kan de HC(T) familie zowel 4mA leveren als opnemen, dus een low power LED van 2mA is nog wel te voeden. Een ouderwetse 20mA LED dan weer niet. En een CD4000 serie kan het met 1mA eigenlijk helemaal niet. Daar moet dus een buffer achter als je die iets anders wilt laten doen dan een poort aansturen. En ook die zijn er weer: bijvoorbeeld een ULN2003. Dat is een 7-voudige buffer die met 1mA in wel 500mA aan kan. Dus als je met digitale schakelingen wil gaan werken, dan moet je voor jezelf nagaan wat voor eisen je gaat stellen aan je bouwstenen: is energieverbruik belangrijk? Of juist snelheid? Of allebei? Verkrijgbaarheid en kosten kunnen ook een rol spelen. Vervolgens kan je met behulp van Google de juiste IC's met bijbehorende datasheets selecteren om de gewenste functie te realiseren. Hopelijk kan je hier wat mee", besloot Opa. Pim keek bedenkelijk. "Tja. Het was weer een berg informatie, maar ik weet nu wel waar ik op letten moet. Bedankt voor de lessen Opa", zei hij, met een schuin oog op het schema waarvan de symbolen voor hem nu in elk geval betekenis hadden gekregen.

De SEM

Wim Kruyf, PA0WV

Inleiding

SEM staat voor SignaalElementMeter van Morse code. Morsecode is tegenwoordig immaterieel cultureel erfgoed, erkend door de UNESCO, dus waag het niet je tengels er naar uit te steken. Hetzelfde geldt voor Tuitjebroekboerenkinkelvoorjaarsgemengdbaltsdansen en voor het in Nederland gemeentelijk verbod op carbidschieten rond elke jaarwisseling; toch gedaan middels 40 liter melkbussen met zuigend passende metalen deksels, dat is wat beschermd wordt; benevens voor de volkssport van het duivenmelken en palletbrandstapels tijdens de jaarwisseling in Scheveningen en Duindorp, zijnde een "prachtwijk" van den Haag.

Goed. Nu is het zo, omdat Morsecode niet meer



verplicht is als onderdeel voor het zendexamen, dat men de voordelen kunnende kennende, (zie dit document) toch wel redelijk vaak de Morsecode wil aanleren teneinde mede aan de classificatie - door 'echte' zendamateurs bedacht - te trachten te voldoen; zijnde dat je bij gebrek aan vaardigheid in het op het gehoor opnemen van Morsecode geen echte zendamateur bent, edoch een geregistreerde CB-er die slechts kan zwetsen op de band over zijn galstenen en zijn problemen in zijn Vogelaar-Prachtwijk met sociale woningbouw en betaalbare huren, om in zijn postzegelachtertuintje tussen de daar geplante en welig tierende cannabisplanten, een mast van 40 m hoog op te mogen richten.

Toppunt van onbenul in mijn optiek van het uitoefenen van het zendamateurisme door volwassen kerels op dergelijk niveau is, als ze met een koopbak en een antenne uit blisterverpakking "kidsday" gaan bedrijven (schijnt geen Nederlands woord voor te zijn) en dat dan blijkt dat een vijfjarige die niet eens kan schrijven, met een melktandjesbekje die koopsets volledig weet uit te nutten.

Maar ja, dan houd je er zo weinig 'echte' over, dat doen we dan maar niet, ten faveure van de koopbakkenlobby.

Je kunt op Internet morse leren opnemen met programma's; online is de meest bekende lcwo.net en offline met G4FON, JustLearnMorsecode, Morsecat en dat soort programma's, die je gratis kunt downloaden. Als je een jaar of 16 bent lukt dat leren in hooguit twee maanden en als je 60 plusser bent kost het een paar jaar om een praktisch bruikbaar niveau te bereiken, het gros knapt dan voortijdig af; goed voorbeeld voor de jeugd....

Het leren seinen wordt echter altijd afgedaan als: "Ach dat leer je zo, als je eenmaal kunt opnemen". Dat is echter duidelijk niet het geval, dat blijkt als je luistert op de banden in de CW gedeelten waar blijkt dat de handschriften van amateurs vaak QSD zijn.

Lees dit artikel eens en daarna dit artikel en je komt er achter hoe vroeger seinen werd geleerd. Nu is 'vroeger' verleden tijd, en tegenwoordig gaat men dan met een morsedecoder zijn eigen seinschrift bekijken. Dat betekent dan vaak: schrikken.

Die decoder-check is overigens een slecht idee, want die werkt met beslissingsniveaus om vast te stellen of een mark een punt of een streep is. Als dat beslissingsniveau halverwege ligt, dus op 2 ditlengte tijdsduur, worden alle marks korter dan 2 dits als punt opgevat en alle marks langer dan 2 dits als een streep. Een dit is hier de tijdsduur van het kortst voorkomende sein-element dus een puntelement of de spatie tussen de elementen van een letter; tekening hieronder verduidelijkt dat.



Je kunt dan dus op het gehoor volledig onneembare Morse maken met punten van 1,9 dit en strepen van 2,1 dit tijdsduur, die op de decoder perfect worden gedecodeerd.

De SEM die hier beschreven wordt meet de verhoudingen in het uit een seinsleutel, keyer, bug of cootie aangeboden schrift (dus snelheids-onafhankelijk) tussen alle tekenelementen, zowel mark als spaces t.o.v. de spaties binnen een letter. De waarden worden op 1 decimaal achter de komma aangegeven.

Het verdient wellicht aanbeveling dat verenigingsafdelingen die morsecursussen organiseren, een dergelijk apparaat nabouwen om de cursisten op te leiden in seinen met een

goed handschrift.

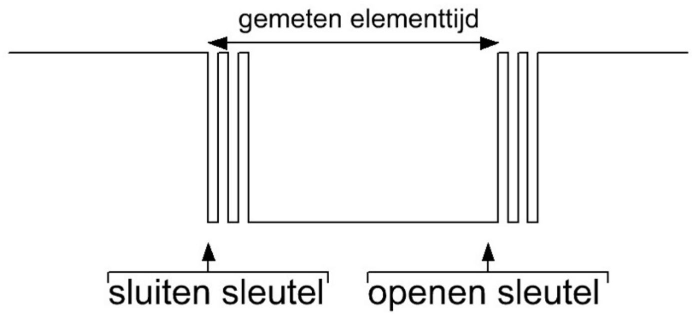
Opzet

Ik ga gebruik maken van een 8 bit micro-controller Atmel AT89S8253, omdat ik die nog voldoende op voorraad heb liggen, en die wordt in assembler geprogrammeerd. Niet gelijk afknappen, je kunt tegen vergoeding van de kostprijs bij mij een geprogrammeerd exemplaar verkrijgen, waardoor nabouw een fluitje van een cent wordt. Ik beschrijf de werking en ik zet er flow charts bij ter toelichting, zodat je met die gegevens ook andere processoren zelf kunt programmeren in andere programmeertalen, zoals bijvoorbeeld een Arduino of een Raspberry Pi, in C en je bovendien begrijpt hoe een en ander werkt.

De elementlengte (tijdsduur) van elke mark en space wordt gemeten en vergeleken met het gemiddelde van de korte spaties binnen letters, om vast te stellen of ze korter dan de grenswaarde zijn of langer. Tevens om te bepalen of het een mark of space was en zo kunnen we 4 verhoudingen bepalen van het gemiddelde van de gesorteerde elementen enerzijds, en het gemiddelde van de korte spaties binnen de letters anderzijds. De verhoudingen worden op een decimaal nauwkeurig bepaald en zodra ze bekend zijn op een display geplaatst en lopend tijdens het seinen bijgehouden.

We bemonsteren het seinsleutelsignaal ongeveer 7200 keer per seconde, en verhogen bij elke bemonstering een tijdsduurteller die maximaal tot FFFF loopt (hexadecimale notatie voor 65535) en daar blijft hangen als die waarde bereikt wordt na ruim 9 seconden omdat opgehouden is met seinen. Als we (te lange) woordspaties willen kunnen meten tot op 9,9 dits lang, komt dit overeen met een minimale seinsnelheid die gemeten kan worden van 1,3 wpm. Dat komt aardig overeen met de met een metronoom toegepaste hiervoor genoemde telmethode.

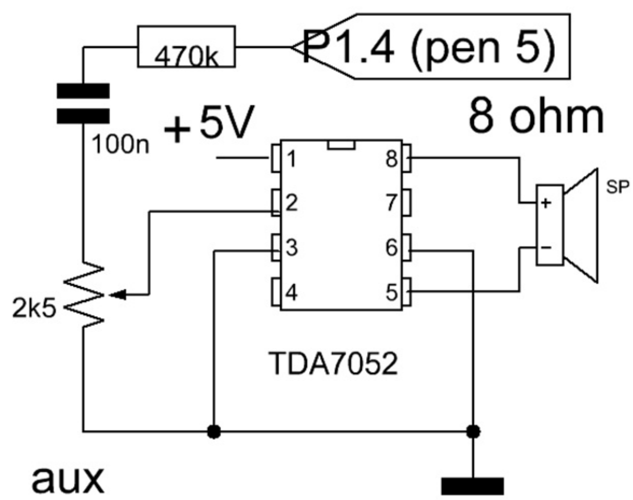
Om bouncing-effecten van de mechanische sleutelcontacten te vermijden, wordt gemeten van de eerste signaalwijziging waarna een dode tijd volgt en wijzigingen worden genegeerd in die dode tijd, tot de eerste wijziging na de dode tijd. Zoals bijstaande figuur ter verduidelijking toont wordt daarmee geen fout en geen vertraging in de meting geïntroduceerd.



Signaalelement met bounce

Het debounced sleutelsignaal wordt uitgevoerd op pen P1.2 van de microprocessor, voor extern gebruik om eventueel sleutelklikvrij een zender te sleutelen, of een decoder aan te sturen.

Inwendig wordt de tweede van de drie beschikbare tellers van de controller gebruikt, die zo is ingesteld dat die 1200 keer per seconde een overflow geeft welke een interrrupt aanroept, om een sidetone in de vorm van een blokgolf op P1.4 van 600 Hz te creëren. In geval een mark wordt ontvangen van de sleutel, inverteert de spanning op die outputpen P1.4 bij elke interrupt, zodat daar dan een blokgolf van 600 Hz op verschijnt. Bij een space vindt de omkering niet plaats en wordt de outputpen laag gehouden. Een audioversterker met een

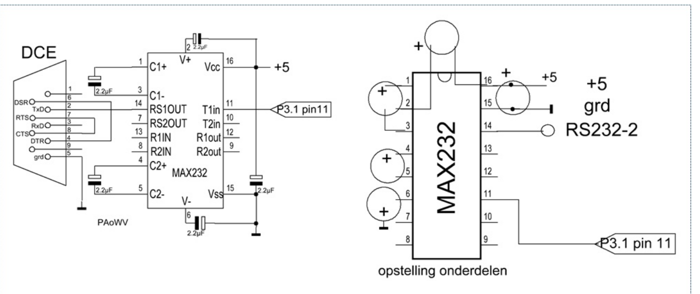


sterkteregeling en een luidsprekertje verzorgen daarmee de sidetone. Met een schakelaar die pen P3.7 naar massa sluit, kan de sidetone worden uitgeschakeld. Dat werkt, maar ik heb de schakelaar niet gemonteerd, omdat op 0 draaien van de sterkteregelaar voor mij ook volstaat.

De samplefrequentie van 7200 wordt gemaakt met de eerste van de drie timer/counters aanwezig in de microcontroller, die steeds tot 255 telt en dan bij de volgende klokpuls terugflopt naar 0; die geeft dan een interrupt af (rond de 7200 keer/s) en dan wordt de sleutelpositie bemonsterd in interruptafhandelingsroutine int1.

De debouncetijd is instelbaar tussen 0 en 255 monsters, dus op maximaal 255 interrups van int1 door teller0 overflow. Met 7200 monsters per seconde is dat dus 35,4 ms. Zo lang is ongewenst, het zou de maximale seinsnelheid trouwens ook beperken tot 33 wpm. Voor een bug is dat aan de krappe kant. Ik heb de debounce-tijd ingesteld op 128 monsters, voor debouncing; royaal en het beperkt de bruikbare snelheid van het apparaat tot ruim 67 wpm. Dat zijn snelheden die je uitsluitend met een keyboard haalt en dan hoef je ook niks te meten want dan zit je sowieso met een elektronisch gegenereerd signaal dat perfecte machinecode is.

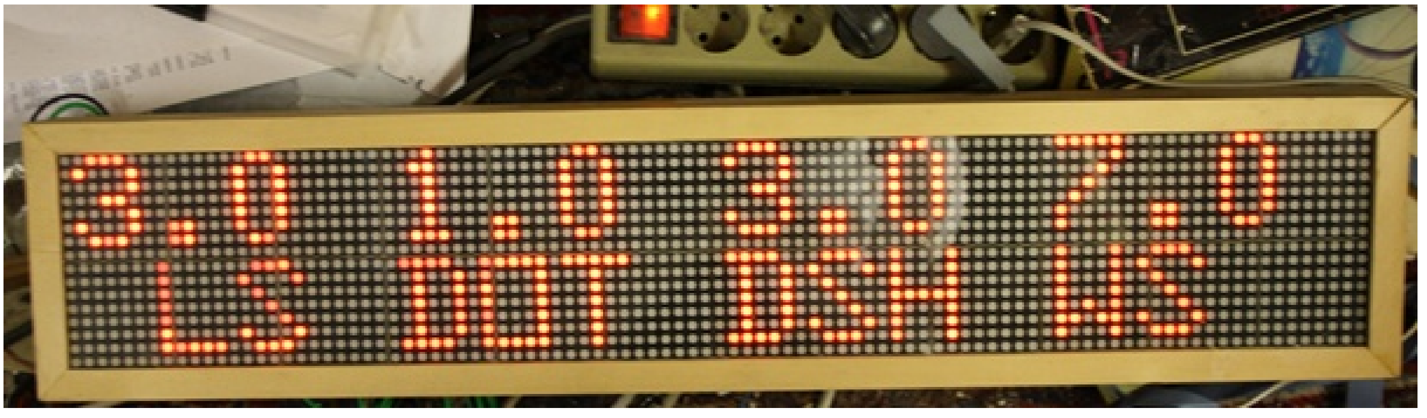
De display van de uit de berekeningen van de gemeten waarden volgende resultaten ga ik doen met een twee maal 16 karakter LCD display, HP44780 compatible. Die gebruikt weinig stroom en kost slechts enkele Euro voor 32 karakters; 7-segment LEDdisplays bieden dan weinig voordeel, en daarvoor is de gebruikte voeding niet toereikend, maar voor demonstratiedoeleinden of klassikaal werken met een morsecursus worden de gegevens via een in de processor ingebouwde UART serieel naar buiten gebracht op een RS232 interface 9 pen female D-connector, en kunnen ze worden aangesloten aan een lichtbak van ruim 75 cm breed, met twee regels van elk 16 karakters, zoals ik die bouwde en destijds twee stuks gebruikte op de dag van de radioamateur zelfbouwtentoonstelling en waarvan ontwerp en bouw beschreven staat op mijn website.



Seriele interface

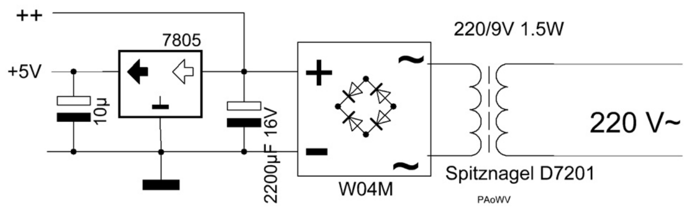
Uitvoering

De ontwerpmethode is de breimethode. Ik schat dat de zaak op een gaatjesbord van half euroformaat kan worden geplaatst (8 bij 10 cm). Printen begin ik niet aan, die zijn bedacht voor massafabricage, en leiden tot onbegrip betreffende zelfbouw door amateurs, waarbij nog komt dat ze experimenten effectief verhinderen en correcte onderdelen van iets andere afmetingen er niet op passen. Op het gaatjesboard eerst een voeding gemonteerd anders kun je lastig iets proberen.



Voeding

In de junkbox vind ik een Spitznagel trafootje primair 220 secundair 2 maal 9 volt waar ik blijkens het stickertje 1 euro voor betaald heb. Dat soort printtrafootjes is inwendig beveiligd tegen oververhitting dus zekeren met een dure 32 mA glaszekering is onnodig. Gelijk twee klodders smeltlijm op de netspannings-soldeer-contacten om opdonders te voorkomen, daar heb ik er al genoeg van gehad sinds mijn elfde levensjaar bij de toenmalige netspanning van 127 volt ter stede.



Voeding

De twee secundaire wikkelingen van 9 volt worden parallel gezet. Niet in serie kortgesloten dus. Een brugcelletje van Baco, een 7805 stabilisator in TO-220 behuizing en een paar elco's completeren het geheel. Even meten of hij 5 volt afgeeft. De hogere spanning, in het schema aangegeven als ++ wordt via een serieweerstand gebruikt om de LEDisplay wat zuinigjes van achtergrondverlichting te voorzien. Bij normale omgevingsverlichting kan dat indien gewenst worden weggelaten, of uitschakelbaar worden gemaakt.

Processor

De processor komt in een 40-pens voet, met als klok een kristal van 11059,2 kHz. De waarde is niet kritisch, 12 MHz kan ook, beiden zijn handelswaarden. De verhoudingen die de display zal aanwijzen blijven hetzelfde maar als ik ook de morsesnelheid in wpm meet en op de display zet, zal die afwijken, net zoveel procent als het kristal afwijkt van de nominale waarde.

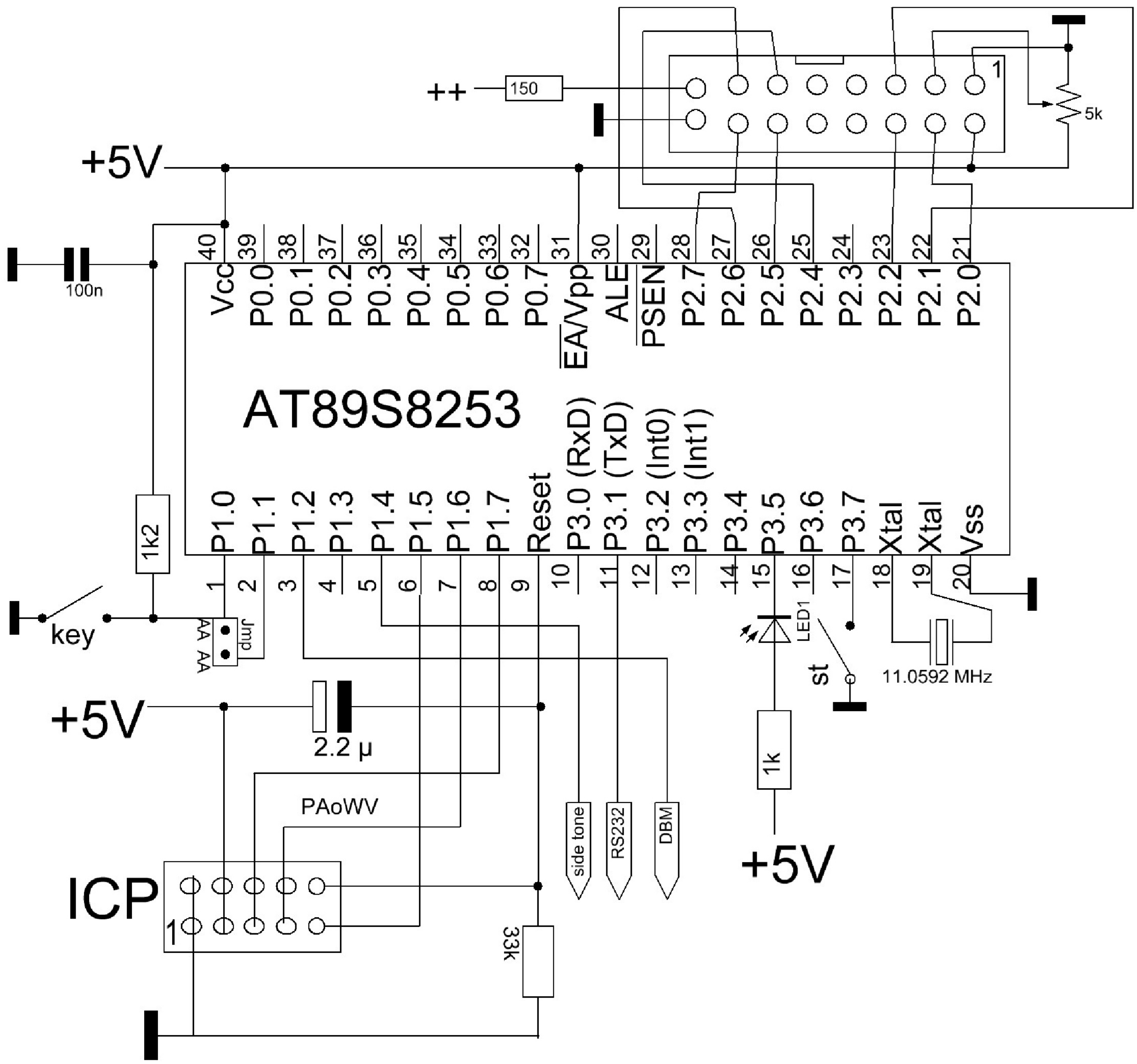
Een 10 pins bandkabelconnector erop geplaatst, zoals het schema aangeeft om hem te kunnen programmeren tijdens het ontwikkelen van de software. Onnodig dus als je een geprogrammeerde chip aanschaft. Meten tussen pen 20 als aarde en pen 31 en pen 40 als plus 5 volt. Als dat inderdaad 5V blijkt, kan de processor in de voet. Opletten dat de pootjes in de gaatjes van de voet gaan en niet onder het IC krombuigen. Tevens opletten dat de processor niet 180 graden gedraaid in de voet wordt gezet. Dan werkt er, als je hem goed erin prikt, zelfs met een ongeprogrammeerde processor, al het nodige, en op pen 30 kan dan eventueel indien gewenst een kanteelspanning met een derde van de kristalfrequentie worden gemeten met een teller, ter controle.

Ontwikkeling software:

Eerst de teller0 activeren en de interruptroutine int1 geprogrammeerd, die de lengte van de signaalelementen met een teller bepaalt, wisselingen van de sleutel debounced en het niveau en de lengte van het signaalelement klaar zet voor ophalen door het hoofdprogramma. Zie de tekening verderop. De cijfers die hier en daar in de flowcharts staan komen overeen met de labelnummering in het assembly-programma.

De figuur bevat ook later in de ontwikkeling aangebrachte modificaties, zoals 2 vlaggen die er voor zorgen dat bij ophouden van seinen, na bijna 10 seconden, als de elementsduurteller FFFF bereikt een vlag wordt gezet die er voor zorgt dat in het hoofdprogramma op de onderste regel van de display de laatst gebruikte morsesnelheid wordt afgedrukt, afgerond op de dichtsbijliggende gehele waarde in woorden per minuut.

Er wordt ook een errorflag gezet als de interrupt de resultaatwaarden van de meting wil updaten maar de eerder geplaatste waarden nog niet zijn opgehaald door het hoofdprogramma. Dat zou dan aanleiding geven op de onderste regel van de display een foutbericht weer te geven. Dat bleek bij testen onnodig. Het hierna te bespreken hoofdprogramma verwerkt de door

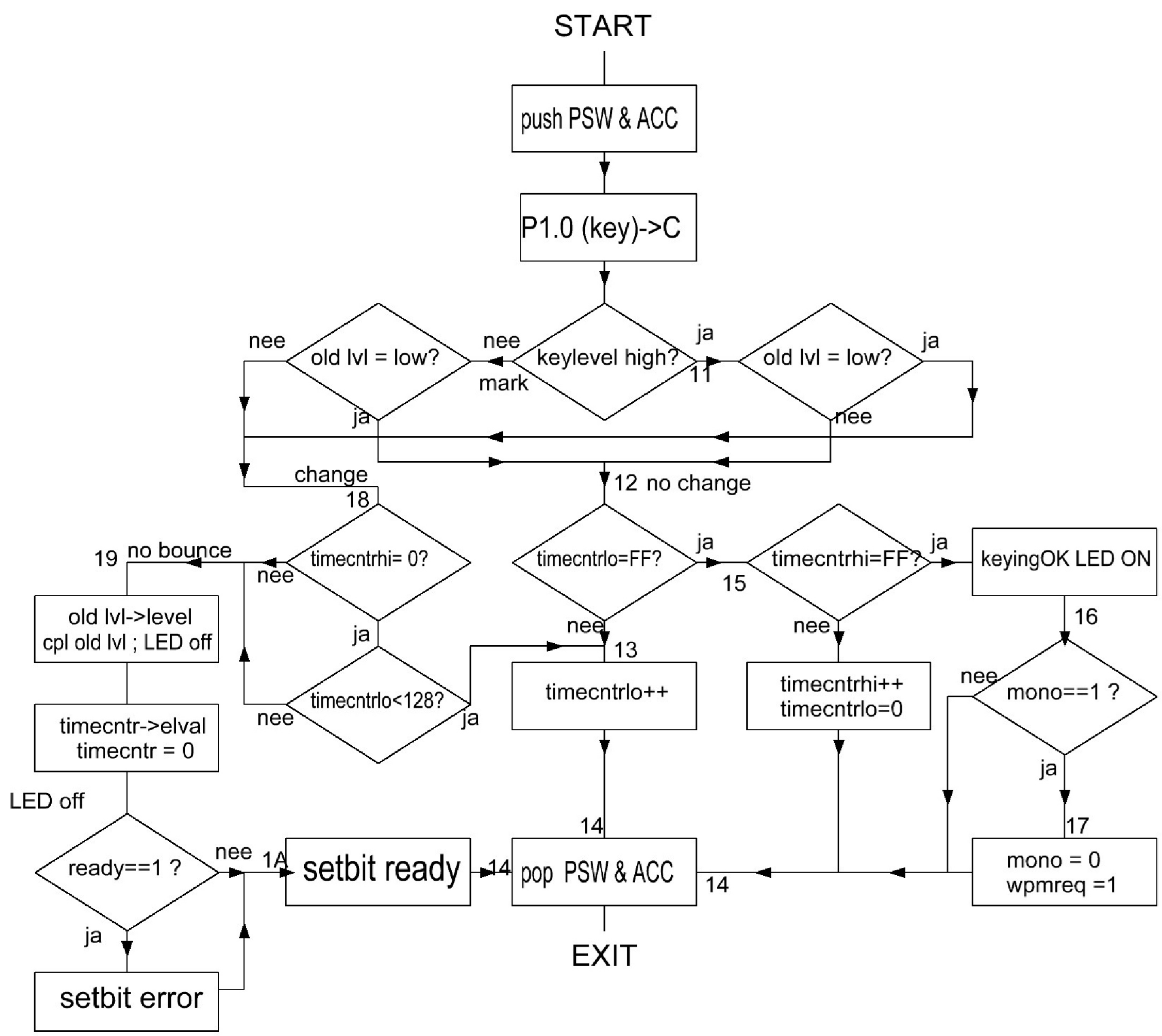


de int1 geleverde signaalelementlengte dus altijd binnen de tijd, zelfs bij zeer hoge snelheden, (120 wpm) waarbij ik tevoren de debouncetijd voor dat experiment had ingekort.

Ook zijn de LCD routines geprogrammeerd zodat er voor debuggen controle mogelijk is om te kijken of een en ander werkt van de gebouwde tussenstukken.

Pen 15 en 16 van de boxed header LCD-voet, via een weerstand van de aangegeven waarde op ++ van de voeding pas aansluiten, als de display blijkt te werken, omdat die correct is aangesloten. Anders loop je kans met de hogere ++ spanning de display op te blazen.

Met de tweede beschikbare teller in 13-bit mode een preset gekozen zodat die 1200 keer per seconde int3 bij overflow aanroept door een interrupt. Versterker en luidsprekertje erachter gehangen en de sidetoon kan gecontroleerd op werking door het sleutelcontact op P1.0 te openen en te sluiten.



int1: debouncing and measuring element duration

Verdere programmering

In het hoofdprogramma 4 stuks circular buffers geprogrammeerd. Een voor korte spaces, een voor lange letter spaces een voor korte en een voor lange marks. Voorlopig maak ik die 7 posities van elk 16 bits breed. Daarmee kunnen we dan de gemiddelde verhouding van het gemiddelde van 7 gelijkwaardige elementen op de display zetten. Bij elke buffer hoort een 24 bits som van de erin geplaatste elementtijden. Per buffer is er een carry bit dat aangeeft of een schrijfoperatie slaagt, zo niet dan is de buffer blijkbaar vol.

Zolang de buffer niet vol is worden de geschreven waarden opgeteld in de som. Is de buffer vol dan wordt het oudste element uitgelezen en afgetrokken van som en het nieuwe element dat er eerst niet in kon, kan er dan wel in en het wordt opgeteld bij som. Zo hebben we steeds de som van alle elementen in de buffer beschikbaar, onafhankelijk van de gekozen lengte van de buffer.

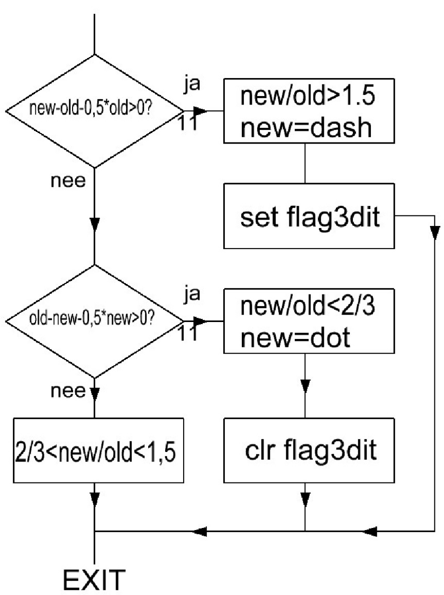
Met die handelwijze bereik je dat som steeds de som van de vier of 7, of zoveel als de buffer

groot is, laatste elementen bevat van een type.

Die som moet dan gedeeld door de som van evenveel referentie-elementen, zijnde de dit-spaties binnen een letter, om de displaywaarden te vinden.

Om te beslissen in welke van de vier buffers een element thuis hoort moeten we vaststellen of het een mark of space is en of het langer of korter dan de beslissingsgrenswaarde is. Indien langer en tevens een space moet ook bekeken of het langer of korter dan 5 dits is. Langer dan 5 dits is een woordspatie, die komen relatief niet vaak voor en daarom worden die zonder buffer voor uitmiddeling onmiddellijk uitgedrukt in dits op de display gezet.

Na een rustpauze hangt de lengteteller op FFFF als spacelengte, die toestand wordt bij het bereiken van die waarde, ongeveer 10 s na afloop van de laatst geseinde mark, gebruikt als seintje om de morsesnelheid te berekenen uit de displaywaarden. Bij starten van het zenden van een nieuwe sessie na die tijd, omdat er weer geseind wordt, wordt die waarde FFFF als waarde van het eerste spatie-element ongebruikt weggegooid. Tevens worden dan de buffers gereset en de display gewist. Soort hot reset dus. De onderste regel bevat dan mnemonic labels die van de erboven komende meetwaarden de betekenis aangeven.



selection: new element is dash dot or unchanged

De flowchart van de verwerking in main staat bovenaan de vorige bladzijde.

Om te bepalen of sprake is van een nominaal 1 of 3-dits element delen we elke waarde van het laatst geleverde seinelement door de voorgaande waarde van dezelfde soort (mark of space). Is dat quotiënt groter dan 2/3 en tevens kleiner dan 3/2 dan is sprake van hetzelfde type element 1 of 3 dits lang, Is echter de breuk groter dan 1,5 of kleiner dan 2/3 tussen twee opeenvolgende door een space gescheiden marks, of twee opeenvolgende door een mark gescheiden spaces, dan is er sprake van een wissel van punt naar streeplengte of van streeplengte naar puntlengte.

Nu lijkt dat lastig vast te stellen met een 8 bits processor geprogrammeerd in assembly voor 16 bits getallen in teller en noemer. Maar dat valt mee. We hebben twee waarden vorige en huidige (old en new in de tekening) die we willen delen en kijken of de waarde >1,5 of <2/3 is

Dat gaat dan met een aftrekking en een shift als volgt in flowchart gezet. Het resultaat wordt in een vlagbit gecodeerd, dat aangeeft of het type element hetzelfde is: de vlag blijft dan ongewijzigd, of dat er sprake is van een kort of lang element dat de vlag dan op 0 of 1 zet.

We wisten uit de interruptroutine al of een waarde een mark of space was, we willen weten of het bij de klasse 1 dit of 3 dits thuishoort om het in een van de 4 buffers erbij te kunnen zetten of, indien langer dan 5 dits space op de wordspace display.

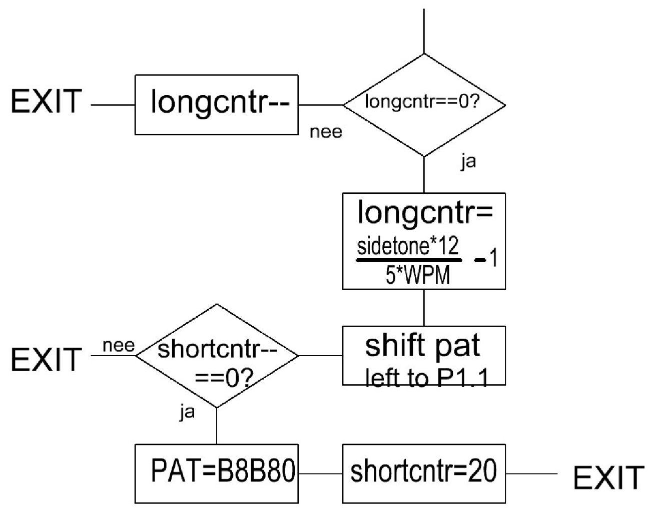
Bij langzaam seinen kom je niet uit met twee bytes voor de som van bufferelementen, dus moet een drie byte deling worden geprogrammeerd voor het bepalen van de verhoudingen. Dat gaat dan met een staartdeling, die bij binaire getallen niet lastig is, omdat de tafels van vermenigvuldiging slechts bestaan uit vier

gevallen in plaats van de tien tafels elk tien lang die je bij decimale getallen in het basisonderwijs hebt geleerd. We delen door tot 4 binaire cijfers achter de komma, binair is dat 1/16, omdat we de verhouding op 1 decimaal achter de komma op de display willen zetten. Omdat de verhouding kleiner is dan 10 volstaan dus ook 4 binaire cijfers voor de kommapositie, zodat het quotient van de deling in een byte past. De omzetting van dat quotient naar decimaal, gebeurt in een aparte routine, die bij het linkse nibble de waarde hex 30 optelt voor het verkrijgen van een decimaal cijfer voor de komma, en voor het rechtse nibble gebeurt de omzetting netjes afgerond op 1 decimaal met een look-up table van 16 posities.

Een en ander is geprogrammeerd en getest en gedebugged middels wat testgevallen die ter beoordeling van de resultaten op de LCD worden gezet. De deelroutine met de staart-deling blijkt tijdens die testen dan bijvoorbeeld 314 microseconden te kosten.

Testsignaal

Voor controle is tijdens de ontwikkeling van de software een test-morsesignaal gewenst, dat wordt als nevenfunctie van de sidetone interrupt afhandelingsroutine int3 gegenereerd op pen P1.1 (pin 2) van de processor. Met een jumper kan het testsignaal op P1.1 met de sleutelinput P1.0 worden verbonden. Een 2 bytes counter zorgt ervoor dat de meeste interrupts geen actie geven, uitsluitend als een dittijd om is is er actie. Een byte volstaat dus net niet want dat zou de snelheid van het testsignaal bij de ingestelde sidetonefrequentie van 600 Hz limiteren op 6 wpm aan de onderzijde. Die actie bestaat uit het shiften van een bit uit een 20 bits register, Daarin is qua code twee maal hex 47 gezet, waardoor er twee letters A worden gegenereerd gevolgd door een woordspatie, en dat almaar door, of er sidetone is ingeschakeld of niet. De juiste werking daarvan is weer gecontroleerd door de erdoor opgewekte sidetone via de laptop microfoon op te nemen met het programma Audacity en daarop de timing te bekijken. De flowchart van deze testroutine staat hieronder getekend.



aa aa generatie op P1.1

Woordspaties

De woordspaties uitgedrukt in interletterspaties vormen een aparte klasse, omdat ze relatief weinig voorkomen worden ze niet uitgemiddeld over zeven stuks maar de laatste wordt op de display gezet.

Dit levert problemen op; we kunnen de woord-spatie slechts uitdrukken in de gemiddelde interletterspatie, die we niet weten want we weten alleen het aantal en de som van de interletterspaties in de buffer. Dat eist dus twee delingen namelijk voor berekenen van de gemiddelde interletterspatie en vervolgens die delen op de gevonden woordspatie.

De display van de 4 te berekenen schrift-parameters werd aanvankelijk elk voor zich slechts gedaan als de 2 buffers die daarvoor per parameter worden gebruikt beiden vol zijn. De scheiding tussen letterspatie en woordspatie kan slechts gebeuren als de grens van 5 dits bekend is. Dat duurt vrij lang omdat de letterspatiebuffer pas kan starten met vullen na ontvangst van 7 interletterspaties. Daarom wordt bijgehouden hoeveel interletterspaties buffer 1 bevat, zodat voor het doel van de scheiding van letter en

woordspaties het vereiste gemiddelde vanaf de ontvangst vanaf een interletterspatie reeds berekend kan worden.

Morsesnelheid

Morsesnelheid wordt uitgedrukt in woorden per minuut, waarbij is afgesproken dat PARIS als standaardwoord daarvoor wordt gebruikt. PARIS bevat inclusief woordspatie 50 dits. Een woord per minuut is dus 50 dits per minuut oftewel 5/6 dit per seconde. Het aantal dits/s wordt ook wel de baudsnelheid genoemd.

Dat geldt uiteraard als alle verhoudingen van het signaal perfect zijn. Wij meten hier echter de imperfectie van signalen. Daarom kunnen we de snelheid meten door voor het woord PARIS de gemeten verhoudingen toe te passen en dan te bepalen op welke snelheid we uitkomen. Maak je bijvoorbeeld de strepen van het signaal te kort dan geeft dat meer woorden per minuut met dezelfde baudsnelheid dan de standaard 1,2 maal de baudsnelheid.

Paris bevat een woordspatie, 4 letterspaties, 4 strepen en 10 dots plus 9 interletterspaties welke laatst genoemde we hier als referentie aanhouden.

De morsesnelheid wordt dan uitgedrukt in onze referentie, zijnde de interletterspaties, van dots, dashes, letterspaties en woordspatie, zoals afleesbaar op de display:

6/5 * 50/{10*(dot/interletterspatie) + 4*(dash/interletterspatie) + 4*(letterspatie/interletterspatie) + (woordspatie/interletterspatie)+9}*(1/dittime)

De laatste factor, 1/dittime, in die formule wordt absoluut gemeten in aantal interrupts per seconde die een dittijd gemiddeld nodig heeft en die berekening is dus afhankelijk van de gebruikte kristalfrequentie. De andere factoren en termen vormen allen verhoudingen die onafhankelijk zijn van de kristalfrequentie. Bij een ander waarde dan de gebruikte kristalfrequentie van 11,0592 MHz klopt de wpm indicatie dus niet, maar die is dan wel terug te rekenen tot de werkelijke waarde gemeten volgens de laatste seintekens, zoals de display vermeldt.

Dit kan berekend worden uit de beschikbare getoonde meetwaarden en geprogrammeerd. voor de gebruikte kristalfrequentie. Voor de verhoudingen tussen de accoladen worden de tijdens het seinen gevonden 1 byte quotienten gebruikt met 4 bits voor en 4 bits achter de komma, die als binair getal opgevat dus 16 keer de quotientwaarde bevatten. Dat wordt gecompenseerd door de teller ook met 16 te vermenigvuldigen.

Het grootste deel van die rekenpartij wordt reeds als berekende constante tijdens het assembleren vastgelegd. Er blijft een deling met 4 bytes in de teller over, die met een aparte deelroutine wordt uitgedeeld. De noemer van die deling is het product van sum1 en de duur van PARIS uitgedrukt in de dits met de geseinde verhoudingen.

Dat product wordt berekend met een apart geprogrammeerde 2 byte maal 4 byte vermenigvuldigroutine. Door een beetje handig te goochelen met dubbel gebruik van geheugenplaatsen is de uitkomst direct op de noemerposities van de 4 byte/ 4 byte deelroutine beschikbaar, en wordt ook ander schaars RAM dubbel gebruikt.

Dat hoeft niet steeds te gebeuren met het daarbij behorende tijdbeslag, maar als je ophoudt met seinen en de elementlengteteller blijft na bijna 10 seconde hangen op FFFF is dat het juiste moment om als toegift je zendsnelheid van je laatst geseinde tekens te berekenen en te tonen op de onderste lijn van de display.

De seinsnelheid wordt alleen berekend als voldoende lang geseind is om alle 4 quotienten op de display verkregen te hebben.

Beneden tien wpm wordt geen 0 maar een spatie op de eerste cijferpositie gezet. Het een-

byte quotient is opgesplitst in 7 cijfers (bits) voor de komma en een erna, zodat de display tot maximaal 127 wpm zou kunnen werken en het cijfer achter de komma, dat decimaal 0 of 0,5 vertegenwoordigt, bepaalt of naar beneden of naar boven wordt afgerond. Dat gaat dan simpel door er 0,5 bij op te tellen en dan gevolgd door een shift naar rechts.

Reset van de buffers, de genoemde hot reset, gebeurt alleen als je na het display van de wpm seinsnelheid weer gaat seinen. Doe je het eerder dan zijn de oude bufferwaarden nog aanwezig en begin je dus met foutieve resultaten, tot alle buffers alleen nieuwe waarden bevatten. Er is daarom een LED onder de display gemonteerd die aangeeft dat je nog niet mag gaan seinen, zolang hij na een seinsessie uit is. Hij dooft weer als je start met een nieuwe seinsessie.

Woordspatie

Uit ervaring met de Fistmeter (zie dit artikel) bij de voormalige Vonkenboerwedstrijden op de jaarlijkse Dag van de Radioamateur weet ik, toen ik behalve een seinsleutel een keer een paddle met keyer meebracht, dat de amateurs die hun test op de paddles uitoefenden een grafiek op hun certificaat kregen die perfecte tijden voor punten, strepen en interletterspaties gaf, maar waarbij de spreiding tussen letters en woordspaties daarmee vergeleken dus opvallend slecht was.

Daarom groot belang voor de woordspaties, vooral omdat die bij het toenemen van de snelheid steeds belangrijker zijn om de morsetekst op het gehoor met decodering in je hoofd, dus zonder meeschrijven of typen, te kunnen blijven volgen.

Woordspatie komt relatief weinig voor, bij Paris bijvoorbeeld maar 1 keer op 28 tekenelementen. Dat was de overweging om geen woordspaties uit te middelen maar direct na detectie uit te drukken in dits en op 1 decimaal achter de komma op de display te zetten.

Dat is een complicatie die is opgelost, door bij elke gedetecteerde lange space die niet direct in de 3 dit space buffer op te bergen maar eerst te kijken of die langer is dan 5 dits. Zo ja naar de wordspace display, zo niet alsnog opbergen in de 3dit space buffer.

Elke nieuw gedetecteerde space die in de klassificatie 3 dit of lang valt, wordt daartoe vermenigvuldigd met de buffersize (aantal elementen dat in de buffer maximaal kan staan) en gedeeld door de lopende som van buffer 1 (de dit spaces buffer) Is de uitkomst van die deling groter dan 5 dan wordt het quotient doorgesluisd naar de display voor word spaces. Zoniet dan wordt de oorspronkelijke waarde aan de 3dit spacebuffer toegevoegd.

UART

De UART is geprogrammeerd voor 9600 Baud 8N1. De grote externe display heeft 2 modi die met een jumper kunnen worden gekozen, te weten de Bugmaster mode en de lineprinter mode.

In de bugmastermode is de bovenste lijn van de display dan een lichtkrant die dus steeds de laatst ontvangen 16 karakters laat zien. De lineprinter mode typt van links naar rechts op de onderste lijn en zet als die vol is de inhoud op de bovenste lijn, wist de onderste lijn en plaatst de cursor links onder.

Die line printer mode gebruik ik, dat houdt in dat bij elke wijziging van de LCD display de gehele display van 32 karakters via de UART op interruptbasis naar de grote externe display wordt gezonden. Dat gebeurt vanuit een 32 byte buffer, nadeel is wel dat bij elke wijziging in de externe display als een bliksemflits in 1/30 seconde de boel opnieuw wordt ingetypt, wat toch een zichtbare flits geeft. Acceptabel maar niet fraai.

Het verdient dus overweging dat niet te doen bij elke nieuwe berekening van een der vier waarden op de externe display maar uitsluitend

als die berekening een andere waarde geeft dan er op die plek stond. Dat is dan ook gerealiseerd. Bij het schrijven in de 32 byte tty buffer, wordt eerst op de schrijfpositie gelezen, en slechts als het gelezen byte ongelijk is aan het te schrijven byte, wordt het geschreven en een dirty flag geset. Bij het berekenen van alle 4 waarden wordt na afloop gekeken of de dirty flag staat en zo ja dan wordt de buffer gedumpt op de grote externe display en de dirty flag gereset. De baudsnelheid aanzienlijk verhogen kan ook, maar dat is niet gedaan omdat mijn uitgangspunt was dat ik die externe display wil gebruiken zoals hij is gepubliceerd (met de jumper verplaatst naar line printer mode)

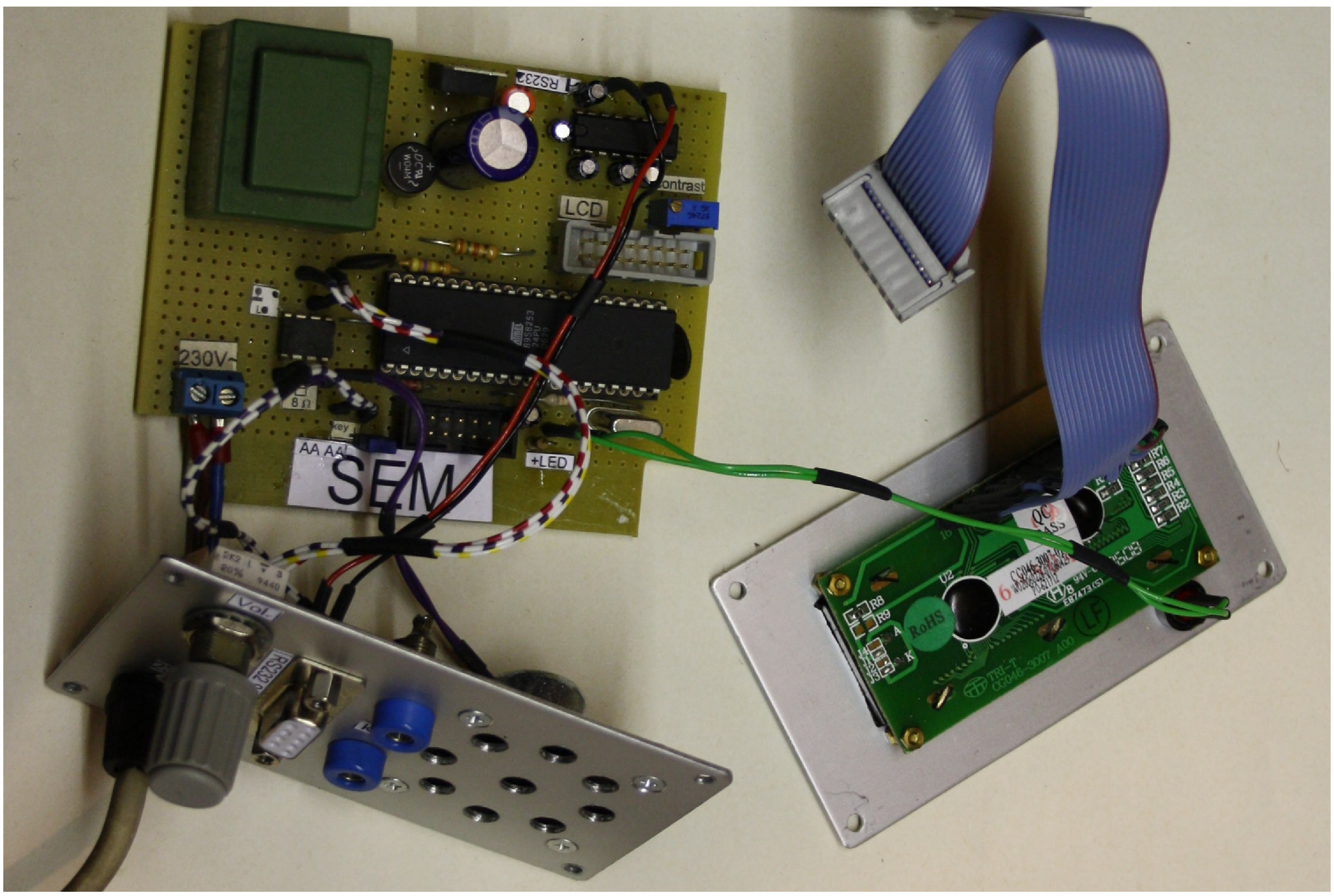
Bij een verversing van de grote display wordt het eerste karakter uit de 32 karakterbuffer in de UART buffer geladen en een idle vlag gereset. Dat is alles, want de uart geeft na elke voltooide letter een interrupt, die de volgende letter laadt, na 32 letters wordt de idle vlag geset, en als bij een volgende interrupt de idle vlag staat wordt er niets geladen en houdt de automatiek dus op.



Nabouw

Foto's tonen de opstelling van de onderdelen. Een geprogrammeerd IC kost 15 euro porto en verpakking voor Nederland en Belgie inbegrepen. Voor bestelling en vragen ben ik bereikbaar per email op mijncall@amsat.org waarbij mijncall vervangen dient te worden door

PA0WV



HF met Handicaps

Het wereldkampioenschap voetbal zit er weer op. Niet dat ik het gevolgd heb, maar ook in de radiohobby kwam je er niet onderuit. Als je zo af en toe op het dx-cluster kijkt (je weet wel, dat stuk software dat je in staat stelt om bijzondere stations te spotten, wat door sommige amateurs nog wel eens verward wordt met het Logbook of the World) dan heb je vast wel reeksen 18-, FWC- en FIFA-stations voorbij zien komen. Die waren allemaal onderdeel van de activiteiten rondom het WK en daarmee waren awards te verdienen. Ik had het veel te laat in de gaten, dus het gouden award ging het niet meer worden. Waarschijnlijk was het dat toch niet geworden, maar dat heeft alles met mijn situatie te maken. En daar wilde ik het eens over hebben, want ik ben natuurlijk niet de enige met beperkingen op hobbygebied.

Want waar ging het om. Om het award te halen, moest je minimaal 500 punten hebben voor brons, 1250 voor zilver en 2018 voor goud. Die punten kon je halen door (veel) speciale stations te werken (die 18, FWC en FIFA stations) op diverse banden in diverse modes. Nou heb ik een paar handicaps, namelijk:

Antenne
Ik heb slechts de beschikking over een inverted-V antenne van 2x 13 meter. Dat is prima voor de banden 30m en lager, maar vanaf 20m moet je eigenlijk een beam hebben om mee te kunnen doen met de grote jongens. Vanwege de lage opstralingshoek, wat DX uiteraard ten goede komt; omdat je een beam meestal wel kunt draaien; en vanwege de gain. Een inverted-V heeft op hoge frequenties een grillig stralings-patroon waardoor je afhankelijk bent van goede condities of er zo'n woeste stralingslob in toevallig de goede richting tegen de ionosfeer wil bonzen. En laten condities er in deze tijd nou net niet veel zijn. Het draaien is handig om de antennebundel in de juiste richting te krijgen. En de gain heb je nodig om gebrek aan vermogen te compenseren (als je je aan de regels houdt tenminste, maar dat doen we, toch?) en om de zwakke signalen nog enigszins op te peppen voor ze de ontvanger bereiken. 100W in een beam met 6dB gain is hetzelfde als 400W in een rondstraler en dat scheelt aan de overkant gewoon een S-punt. Die heb je in een pile-up hard nodig. (Het scheelt je aan de ontvangstkant trouwens ook een S-punt, en ook dat helpt)

Vermogen
Nog zo'n handicap. Ik heb geen N-machtiging dus ook geen lineair. Mijn enige zonde is mijn gereviseerde FT101, die in de pieken 150-180W kan maken, afhankelijk van de gekozen band. Net even meer dan mag, maar nog geen 3dB. Braaf, niet? Maar in veel landen om ons heen is onze 100W het stuurvermogen voor de eindtrap. Daar wordt makkelijk tussen de 800 en 1500W de lucht in geblazen, dus lig je daarmee zo'n 2 S-punten achter. Het is niet anders.

Modes
Lang niet iedereen beheerst alle modes, en daarmee verlies je een hele reeks aan mogelijke QSO's. Ik beheers redelijk CW, en met de genoemde handicaps is dat nou net weer een enorm voordeel. Ten eerste heb je dan na een uur niet een rauwe strot, ten tweede krijg je minder snel last met de buren en/of de XYL vanwege je onzinnige geschreeuw (in de oren van een leek) waarmee je de afstand tot je tegenstation zonder radio probeert te ovebruggen, en last but not least maak je met CW meer kans om gehoord te worden dan in phone, maar daarover later meer. Mijn handicapmode is de digi mode. Meervoud, eigenlijk: digimodes. Ik vind computer-verbindingen geen QSO's en die gebruik ik dan ook niet, waarmee ik mezelf best wat beperkingen opleg. Ongeveer een derde van het aantal mogelijke QSO's, feitelijk. En als je geen CW spreekt, wordt het nóg beperkter.

Ga er maar aan staan. Je mist een mode of wat, en door de antennesituatie en het beschikbare vermogen lig je ook nog eens 3 S-punten achter op de concurrentie. Heeft het dan nog wel zin om aan dit soort dingen mee te doen, vraag je je dan af.

Dat hangt er vanaf hoe je er in staat. Als je alleen voor het allerhoogste gaat, dan gaat het ‘m niet worden. Maar dan heb je waarschijnlijk toch wel die 5 elements multiband beam op een 50m hoge mast plus dikke lineair. Wil je gewoon de lol beleven van toch dat station kunnen werken in de pile-up, en aan het einde nog een award meepakken ook al is het geen goud, dan heeft het zeker zin. En daarvoor geef ik nu een aantal van mijn trucs weg...

CW tips

Ik maak om voornoemde redenen meestal gebruik van CW. Daarmee heb ik de meeste kansen om door de pile-up te breken. Hoe doe ik dat?

● Luisteren. (moet je altijd, toch?). Wat is de modus operandus van het DX-station. Gaat hij meteen over op luisteren na zijn TU, geeft hij eerst nog zijn call, hoe vaak doet hij dat - allemaal dingen die belangrijk zijn voor je timing, ofwel het moment waarop jij je call geeft. Sommige stations geven hun call na elke verbinding, soms zelfs twee keer (niet alle special event stations zijn doorgewinterde expeditie operators), soms om de paar QSO’s en sommigen zelfs minutenlang niet, wat dan weer questionmarks oplevert van stations die willen weten welk station actief is. Wat het DX-station vervolgens niet hoort omdat hij spit werkt… Luister dus ook eens wat hoger in de band of dat het geval is.

● Split werken. Ik werk bijna altijd split, ook al werkt het DX-station simplex. In het geval dat het DX-station simplex werkt zet ik mijn zendfrequentie 50-150Hz boven zijn werkfrequentie, en soms eronder. Waarom? Als het DX-station zijn CQ, zijn call of TU heeft gegeven, breekt de pleuris uit. Iedereen begint zijn call te seinen. Uit onze Liechtenstein ervaring weten we dat dit dan klinkt als een constante fluittoon, veroorzaakt door alle tegenstations tegelijk. Het toontje wat dan afwijkt in frequentie valt dan op en zo kan je je onderscheiden, ook al ben je zachter dan de rest.

● Werkt het DX-station split (aangegeven doordat het DX-station UP seint na zijn call, al dan niet gevolgd door een 1), dan ga ik 800Hz boven zijn frequentie zitten en niet precies 1kHz, ook weer om op te vallen. Vooral als hij door de band draait, helpt het om de ondergrens op te zoeken.

● En als hij split door de band draait, wissel dan regelmatig van VFO om te zien wat zijn grenzen zijn. Blijft hij +1 luisteren? Of varieert hij van +1 tot +3 bijvoorbeeld. En waar is hij op dat moment? Op de lage banden hoor je vaak ook zijn tegenstation, waardoor je weet hoe hoog hij luistert en je meteen terug kunt komen op de laatst gebruikte frequentie. Het slechte nieuws is dat de meeste geroutineerde DX-jagers dit spelletje natuurlijk ook kennen.

● Timing. Zoals ik eerder aangaf, is dat belangrijk. De strijd meteen na de over verlies je. Dat is het moment waarop de kanonnen met beams en lineairs dwars over je heen gaan. Pikt hij er daar een van uit, dan wacht je rustig af. Na een paar minuten zijn de kanonnen weg en levelt het speelveld. Dat is het moment dat de fluittoon zijn werk doet en hij die eerste seconden na de over niets meer kan onderscheiden. Na die paar seconden geef jij je call, als het net even stil wordt. En ziedaar, hij is voor jou.

● Straight key gebruiken. Bloedirritant, maar soms werkt het. Bijna alle DX-jagers, evenals het DX-station zelf, zitten meestal tegen de 30 woorden per minuut. Als jij met je handpomp op 15wpm je call gaat zitten geven, loopt je call qua tijd dus door in de uitloop na de fluittoon en krijgt hij je rustig geseinde suffix nog mee. Gevolg: “CNO?” En daarna kan je op je gemak je hele call geven. Als een ander het doet, vind ik het niet leuk, want zo’n slak haalt heel het ritme uit de pile-up, maar in sommige gevallen werkt het wel, al was het maar omdat het DX-station van die slak af wil die zijn QSO-rate (het aantal QSO’s per uur) om zeep helpt…

● Kijk eens om je heen. Heel vaak ga je af op het DX-cluster om een station te vinden, maar vooral bij dit soort evenementen waarbij meerdere stations actief zijn, blijken een paar kHz boven of onder het gemelde station nog andere stations uit de reeks te zitten, die nog niet op het cluster staan. En die zijn dan heel gemakkelijk te werken omdat de hele wereld er nog niet bovenop zit.

SSB tips

Nou begeef ik me op glad ijs, want ik gebruik SSB bij voorkeur niet. Nog afgezien van het omgevingslawaai dat schreeuwen in een microfoon met zich meebrengt, zijn de winkansen gewoon veel minder met beperkte middelen. Met de genoemde 3 S-punten verlies ten opzichte van de grote jongens is het best een uitdaging om gehoord te worden. En hier helpt het niet om 50-150Hz hoger of lager te gaan zitten, want dan word je effectief niet verstaan. Maar er zijn wel een paar dingen die je kunt doen.

● Luisteren. Alweer. Voor je timing is het gewoon belangrijk om te weten hoe het DX-station werkt, en daar kom je alleen maar achter door een tijdje te luisteren. Zo kom je er ook achter of het DX-station split werkt, zodat je niet de bandpolitie over je heen krijgt (UP! UP!) als je meteen begint te blèren op de frequentie van het DX-station.

● Gebruik een compressor, als je die hebt. Daardoor stijgt de gemiddelde energieinhoud van je signaal en dat helpt in een pile-up. Overdrijf het niet, want dan word je onverstaanbaar. In de pile-up zat een 9A station met overduidelijk een niet goed ingestelde lineair. Hij was op de werkfrequentie S9+40dB op mijn meter, ondanks de matige condities op die band, maar was tientallen kiloHerzen links en rechts van de frequentie te horen met zijn splatter. Daar doe je mede-amateurs geen plezier mee.

● Laat die eerste seconden na de over weer aan je voorbijgaan. Je bent in een pile-up geen partij voor 7 elements monobanders met 1500W. Tenzij de condities toevallig heel goed zijn jouw kant op, wat te zien is aan S9+20 signalen op je ontvanger. Geef je call pas na een paar seconden. Haalt hij de kanonnen er eerst uit, wacht tot die afgehandeld zijn.

● Geef je call langgerekt. In mijn geval: Papa Alfa three Charlieeeeee Novembeeeeeeeer Oscaaaaaaaaar. Dat valt net even meer op dan kort je call geven.

● Geef altijd je hele call. Probeer niet met een suffix alleen de oorlog te winnen. Als ik aan de andere kant van de pile-up zit in Liechtenstein, laat ik dat soort stations altijd links liggen. Het kost operators gewoon teveel tijd om eerst de suffix aan te roepen, om daarna de hele call pas te krijgen, die je dan nog een keer moet herhalen ter bevestiging voor je verder kunt met de volgende. Ervaren operators negeren suffixers om deze reden.

● Soms helpt het om af te wijken van het NATO alfabet. Vooral de Oscar lijkt om een of andere reden door zuidelijke landen niet goed verstaan te worden. Hoewel, door Nederlanders ook niet, getuige het feit dat PI4KAR AH18FWC op het cluster zette. Volgens het principe “Work first, worry later” draaide ik meteen naar 18.130 en riep hem aan (het was nog rustig) om er vervolgens achter te komen dat het AO18FWC was, die ik al had op die band en in die mode. Maar ik heb in SSB ook een paar keer gehad dat ik als PA3CNH verstaan werd om voor mij onduidelijke redenen. Daarom gebruik ik soms Ontario of Ocean voor de O. Over het algemeen is het af te raden om af te wijken van het NATO alfabet omdat het er niet duidelijker van wordt met de meest kromme woorden die een letter voor moeten stellen en soms ook niet te verstaan zijn, maar in sommige gevallen helpt het dus. Ik gebruik dan Canada November Ontario. En krijg dan vervolgens als bevestiging Charlie Nancy Ocean terug…

● Split werken. In SSB begint dat vaak pas bij +5kHz. Blijft hij +5? Of luistert hij ook hoger? Soms gaan ze zelfs tot +15kHz. Probeer een gat te vinden waar het nog niet zo druk is, als je de tegenstations kunt horen tenminste. De meeste tegenstations blijven op +5 hangen: een stukje opschuiven (+8) kan net het verschil maken.

Voor beide modes geldt: heb geduld. Het kan je

zomaar een kwartier of meer kosten om één station te werken. In een aantal gevallen waren de stations op mijn ontvanger maar net boven de ruis, maar toch konden ze mij horen. En soms willen de condities gewoon niet meewerken. Rusland op 160, 80 en 40 is overdag met de huidige condities geen optie. 30m werkte met wisselend succes, maar vaak diep in de ruis met trage QSB. 20 en 17m gingen eigenlijk nog het best. En op 15, 12 en 10m hoorde ik meestal niets. Althans, niet het station dat ik wilde werken. Wel allerlei andere stations, maar dat leverde geen punten op. Die banden zijn met een inverted-V bijna niet te doen. Desondanks heb ik er toch nog een paar verbindingen kunnen maken, maar dat is vooral te danken aan de goede antennes en oren van het tegenstation. Met deze beperkte middelen zijn operating practice en geduld de beste ingrediënten voor succes. Laat je dus niet afschrikken door je ongunstige antenne(situatie), maar ga ervoor. De voldoening is des te groter. In mijn geval door toch nog op de laatste dag voldoende punten te verzamelen voor het bronzen award. Mijn aantal gewerkte stations per band zie je in het overzicht hieronder.

						8	1		9		24	12		10	3		2	2					1		
CW	PH	DIG	CW	PH	DIG	CW	PH	DIG	CW	DIG	CW	PH	DIG	CW	PH	DIG	CW	PH	DIG	CW	PH	DIG	CW	PH	DIG
160M			**80M**			**40M**			**30M**		**20M**			**17M**			**15M**			**12M**			**10M**		

20m Loop antenne

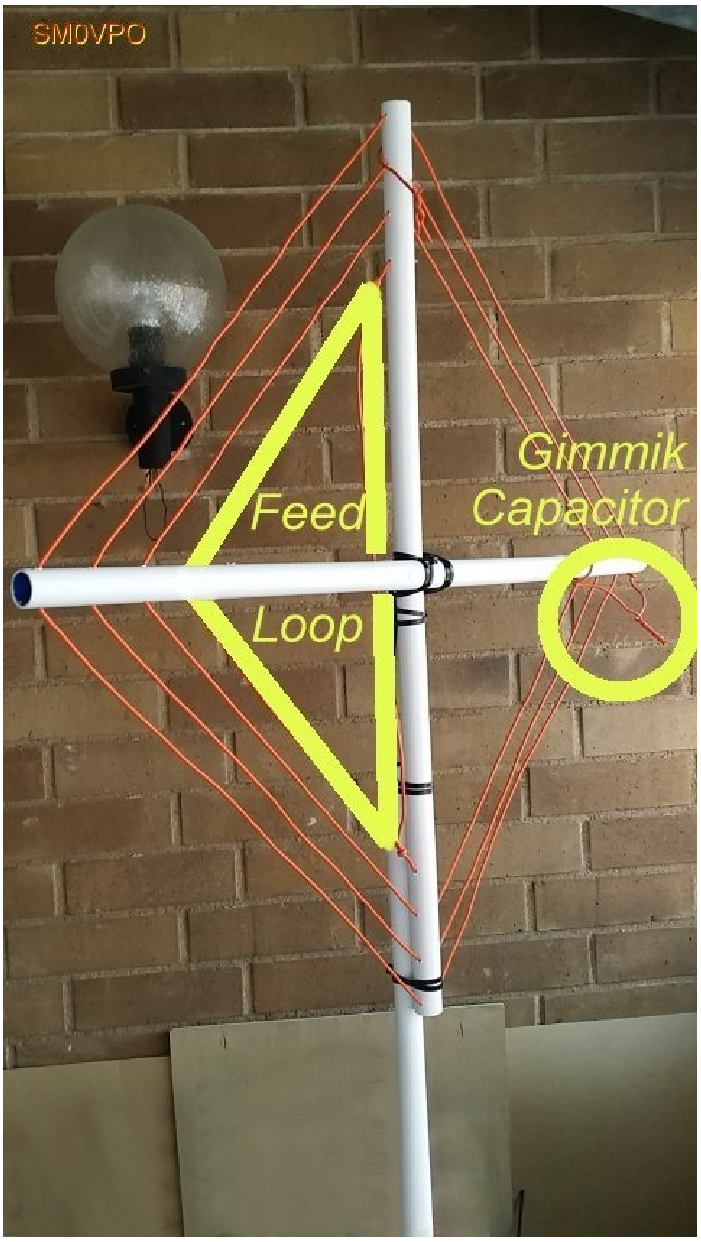
Loop antennes zijn er in allerlei soorten en maten, en als zodanig al vaker beschreven in de RAZzies, maar hier is er eentje die specifiek voor de 20m band is, gering in afmetingen, bijna niets kost aan materialen en niet moeilijk te maken is. Wat wil een amateur nog meer. De antenne is bedacht door Harry Lythall, SM0VPO, die eerder al een 80m versie maakte en nu voor zijn bescheiden appartementje een antenne nodig had waarmee hij op 20m uit kon komen zonder meteen alle buren op zijn nek te krijgen. Zijn eisenpakket was als volgt:

- Zo efficiënt mogelijk (bruikbaar)
- Klein, en draagbaar, zodat hij ook in het veld gebruikt kan worden
- Geen dure componenten, alles lokaal te krijgen
- Geen TVI, QRM of storing aan stereo's of computer geluid
- Totale prijs onder de €2

Harry ritselde bij een installateur op zijn werk wat stukken 5/8" installatiebuis (15mm) van 80cm lang, en een restant 2,5mm^2 soepel installatiedraad (dus met meerdere aders). Meer had hij niet nodig. Hij ging nog in zijn junkbox op zoek naar een afstemcondensator, maar die had hij niet meer. Maar wie heeft een afstemcondensator nodig als je de loop alleen maar voor 1 band gebruikt. Als de loop eenmaal afgestemd is, hoef je dat als het goed is nooit meer te doen. Stem je de loop af op 14.175 (SSB-ers..) en zorg je dat de Q van de loop rond de 100 ligt, dan zou de 3dB bandbreedte meer dan 150kHz moeten zijn. En dat levert een bereik op van 14.100 tot 14.250MHz. Draai als condensator gewoon twee stukken draad in elkaar en knip er zoveel af dat je op de gewenste resonantiefrequentie uitkomt. Dan moet je de loop zo maken dat je voldoende draadlengte en eigencapaciteit hebt om de resonantiefrequentie in de buurt van de 14.5MHz uit te laten komen zonder extra capaciteit. Vervolgens heb je dan nog maar een paar pF nodig om de zaak op frequentie te krijgen en dat lukt wel met een paar stukken in elkaar gedraaid draad.

Constructie

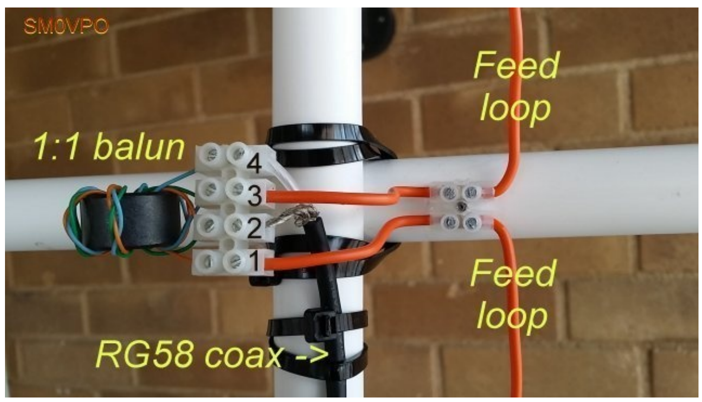
Na veel experimenteren bleek dat precies 3



windingen met een tussenruimte van 2,5cm een eigenresonantie van ongeveer 14,9 MHz oplevert. De gaten waar de draad doorheen gestoken wordt, zitten precies 4cm uit elkaar, beginnend op 1cm vanaf het einde van elke buis. De twee buizen worden in een X-vorm aan elkaar bevestigd met kabelbinders (ty-raps). De voedingsloop is 1/2 winding. Zie ook de foto hierboven. De loop begint rechts aan de buitenkant, gaat dan naar boven, dan linksaf en uiteindelijk eindig je weer rechts door het derde gaatje vanaf de rand. Harry legde knopen in de draad en liet voldoende draad over om de condensator te maken door de uiteinden van de draden in elkaar te draaien. Om de voedings-loop op zijn plaats te houden, heeft hij zo te zien binnendoor de buis nog een stuk draad gespannen waarmee hij de eigenlijke loop ondersteunt op de hoeken. Ik zou dat met touw doen, om beïnvloeding door de draad in de buis te voorkomen.

Het eerste ontwerp had het probleem dat er HF terug kwam over de buitenmantel van de coax. En als Harry dan zijn oude FT-101ZD gebruikte, kon je het HF op de mike voelen met je lippen... De oplossing daarvoor was een gebalanceerde aansturing met tenminste 5m HF kabel.

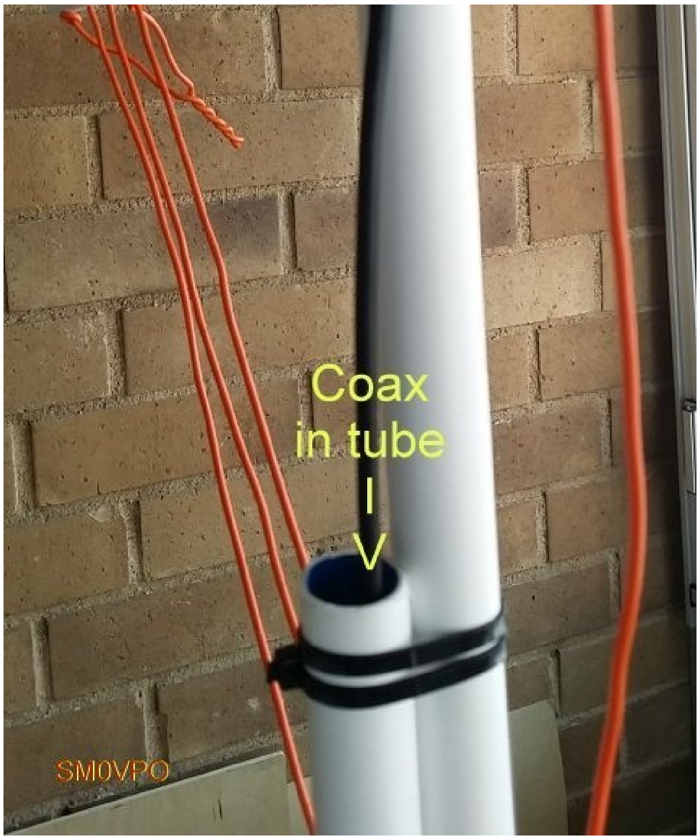
Voor de balun haalde hij een ferrietring uit een oude ATX computervoeding en daarvan maakte hij een trifilair gewikkelde transformator. Hoe doe je dat: draai 3 draden bestaand uit 1mm soepel montagedraad in elkaar (met 7 adertjes als geleider). Leg hiermee 7 windingen op de ringkern en schakel de drie windingen in serie, waardoor je vier aansluitingen krijgt. De aanslui-tingen aan de antennekant zijn genummerd als 1 en 3 en gaan naar de antenne voedingsloop (zie foto). De buitenmantel van de coax gaat naar aansluiting 2, en de middenader van de coax gaat naar aansluting 4. De balun blijft gewoon hangen aan zijn draden en heeft geen extra bevestiging.



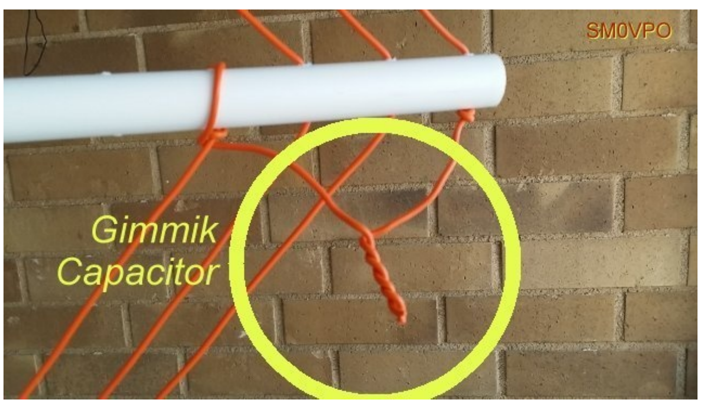
De coax voedingskabel bleek de resonantie-frequentie licht te beïnvloeden, dus daarom wordt deze door een extra stuk installatiepijp geleid zodat hij op zijn plaats blijft, zie de foto op de volgende bladzijde. En dat werkt prima.

Testen

Het testen van de loop is heel eenvoudig. Harry gebruikte een Griddipper om de middelste winding van de loop te controleren op een dip.



Draai de uiteinden van de loop in elkaar zodat ze een condensator vormen en zet de centrale frequentie op 14.175MHz. Met een griddipper kan je dat met een nauwkeurigheid van 100kHz tot 200kHz wel voor elkaar krijgen, maar je kunt ook de SWR controleren met je HF radio. Daarmee kan je trouwens ook door de band draaien en het punt vinden waar de ruis maximaal is: dan komt je dicht genoeg in de buurt.



De centrale frequentie van Harry's 20m loop antenne is 14.175MHz, en de SWR is beter dan 1.05:1 (de SWR-meter beweegt nauwelijks). De Q-factor ligt ergens tegen de 100. De bruikbare bandbreedte is net iets smaller dan hij had gewenst, maar de antenne werkt zeker goed en voldoet aan alle andere criteria. Maar het iets minder bruikbare bandbreedtecriterium gaat ten koste van betere prestaties en het staat hem nog steeds toe om 14.070 MHz te gebruiken, hoewel het daar wat rustiger is. Hoe het verloop van de SWR is in de band zie je in de tabel hiernaast. Effectief dus 14.095 - 14.260 MHz.

Frequency	VSWR
14.010	3:1
14.095	2:1
14.130	1.5:1
14.175	1:1
14.220	1.5:1
14.260	2:1
14.325	3:1



Hierboven zie je een foto van de complete antenne. De "condensator" zie je rechts, en de voedingsloop links. De coax wordt door de steunbuis gevoerd.

QRO uitbreiding

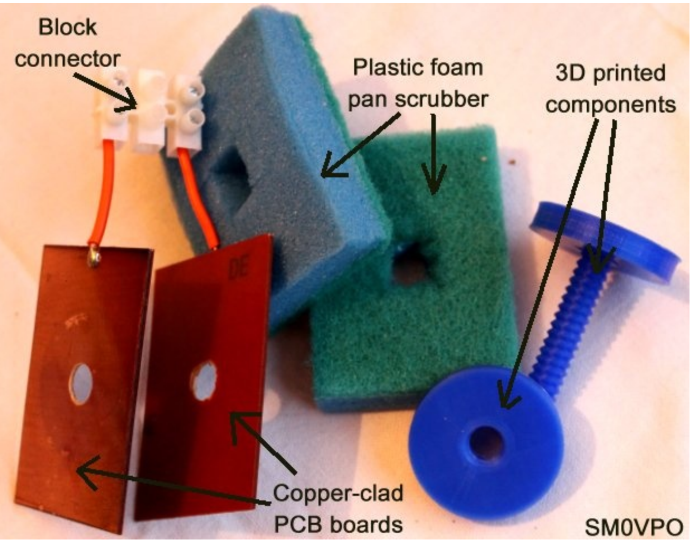
Deze constructie werkt heel goed voor QRP, waarbij het gemiddelde HF vermogen 5 Watt of minder is. Stook je meer vermogen, dan neigt de condensator met de in elkaar gedraaide draden warm te worden, en verschuift de afstemming iets. Maar als je meer dan ongeveer 10 Watt continu HF vermogen erin stopt, dan gaan er dingen in de fik. Goedkoop geïsoleerd draad gaat ook nog roken. Dat komt omdat de impedantie aan het uiteinde van de spoel zo hoog is dat er vele honderden Volts (tot een paar duizend Volt) overheen komt te staan en de isolatie, die voor normaal huis- tuin- en keukengebruik is, doorslaat (en er overslag ontstaat aan de uiteinden van de in elkaar gedraaide draden).

Maar daar is een oplossing voor. In deze paragraaf wordt een super goedkope afstem-condensator besproken die tot 10.000 Volt HF aan kan waardoor je ongeveer 100 Watt HF in deze 14MHz (20m) antenne kunt stoppen. De condensator is instelbaar zodat je 'm met de hand af kunt stemmen (nadat je het HF uitge-schakeld hebt natuurlijk :-). Het afstembereik loopt vanaf ongeveer 8pF tot dik over de 30pF als hij flink ingedraaid is. Het normale bereik voor de antenne is ongeveer 12pF tot 15pF.

Opbouw

Het prototype oogt niet erg fraai, en wordt uitsluitend op zijn plek gehouden door de draden naar de antenne. Er worden slechts een paar componenten gebruikt voor de bouw van de 10kV afstemcondensator:

- twee plasticschuim pannensponzen of één dikke plastic badspons
- twee stukken metaal van 4cm x 6cm. Printplaat werkt prima.
- één plastic bout en moer - zie tekst
- één stevige kroonsteen met 3 aansluitingen waarvan de middenaansluiting verwijderd wordt



Componenten voor de afstemcondensator

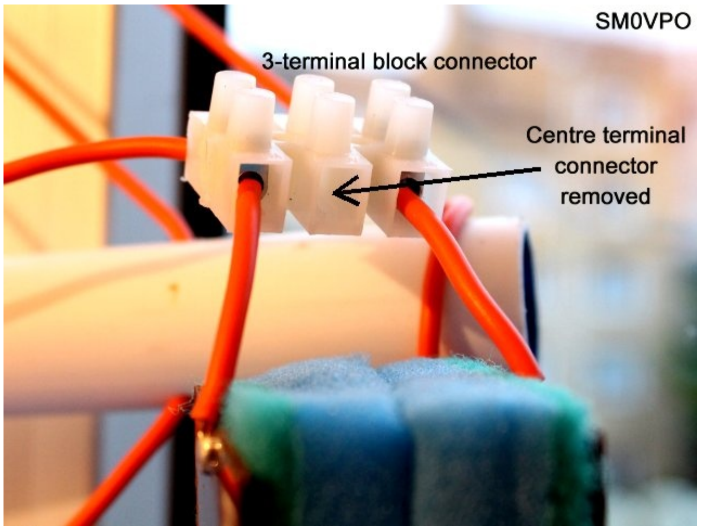
De plastic bout moet ongeveer 6cm lang zijn en absoluut niet geleidend zijn. Maak je de bout zelf met een 3D printer, gebruik dan GEEN zwart plastic aangezien sommige plastics koolstof als kleurstof gebruiken. Je kunt ook een nylon bout gebruiken die IKEA levert voor het bevestigen van WC-brillen, maar je hebt er wel een ring bij nodig om de druk te verdelen. Anders heeft printplaat de neiging om krom te gaan trekken onder de druk. In het prototype is printplaat gebruikt omdat het makkelijk te solderen is - geen noodzaak om gaten te boren en bevestigingsboutjes erin te schroeven. Het prototype maakte gebruik van een met een 3D printer gefabriceerde bout en moer. De benodigde bestanden daarvoor staan onder dit artikel.



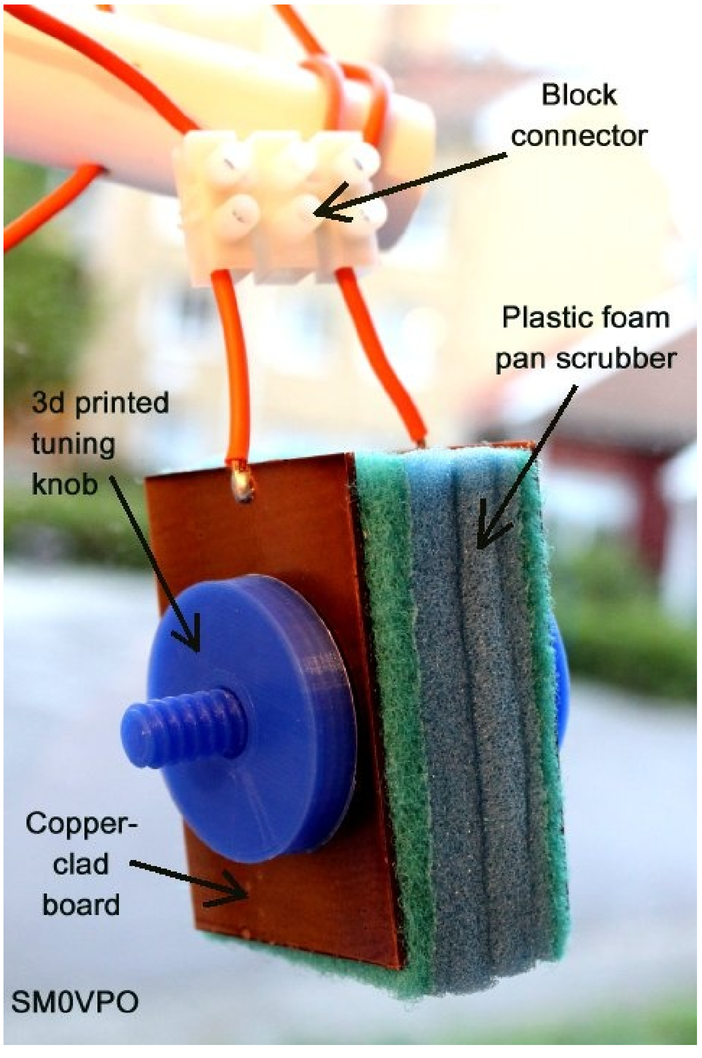
IKEA WC-bouten

Let erop dat de aansluitdraden van de condensator ver genoeg uit elkaar gehouden moeten worden om 10kV te kunnen weerstaan. Daarom is er een 3-voudige kroonsteen gebruikt

waar het metaal van de middelste verbinding is weggehaald. Bij 100 Watt continu loopt de temperatuur na een paar minuten iets op, maar geen tekenen van rook, vonken of puntontladingen :-)

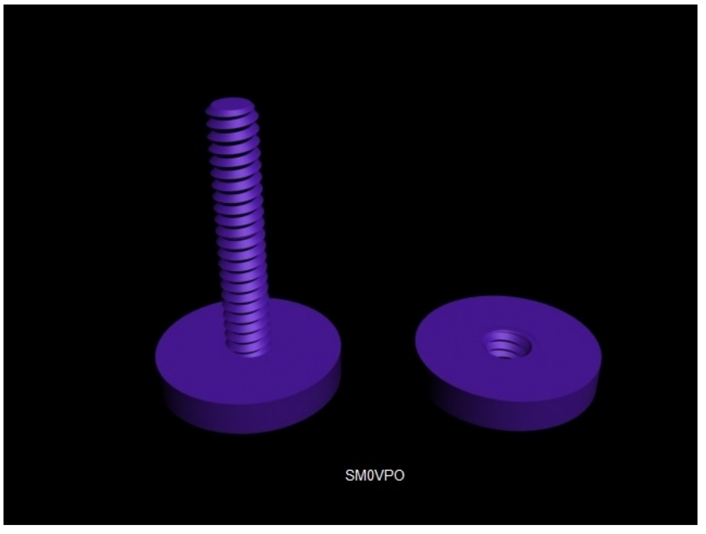


Er zijn diverse types plastic diëlectricum geprobeerd en allen werkten ze goed, mits 100% droog. De beste kwamen nog uit een 10-



stuks verpakking pannensponzen dat €1,50 kostte. Op de foto linksonder de pagina zie je de afstemcondensator met 3D-geprinte bout en moer.

Nogmaals, de condensator is heel eenvoudig te maken met een 3D printer. Het moeilijkste stuk was nog om de schroefdraad goed te krijgen, zodanig dat het over grotere lengtes goed werkte. De bout is vertikaal geprint, zodat de hoek onder de schoefdraad steiler is dan erboven. Dat maakt het printen een stuk makkelijker, Is de hoek te groot, dan wordt het materiaal in de lucht gespoten.... De bout en moer in het prototype zijn ongeveert 3cm in diameter, en de 10mm schroefdraad voor de moer is gesneden met gebruik van boolean subtraction. De moer werd daarna 3% vergroot zodat hij nog wel past maar niet klemvast komt te zitten. Misschien iets overdreven, maar het werkt.



Hier zijn de bestanden:
3D studio MAX file - 3d-cap-01.max
ASCII STL file - 3d-cap-01.stl
ASCII OBJ file - 3d-cap-01.obj
De GCODE file voor de Wanhao (Prusa) Duplicator i3

De printer instellingen in deze GCODE zijn:

- Nozzle temperature = 200°C
- Bed temperature = 60°C
- Support structure = brim
- Layer height = 0.1mm
- Print speed = 60mm/s
- Fill density = 40%
- Shell thickness = 1.2mm

De printer temperatuur is 200°C, en dat is 5°C warmer dan aanbevolen voor PLA materiaal. De ervaring is dat het printen beter gaat bij 200°C omdat het dan een stuk beter blijft plakken. Bij het printen van de schroefdraad is het advies om te beginnen met printen op 200°C en na een paar geprinte lagen de temperatuur terug te brengen naar 195°C.

Noot van de redactie: zelfs als je geen handige knutselaar bent of niet over een 3D printer beschikt, kan je deze antenne nog bouwen. Dan gebruik je gewoon de IKEA wc-bril bouten met een vulring (die je desnoods ook weer van -dik- printplaat kunt maken). Loop antennes zijn vrijwel altijd een compromis vergeleken met een dipool. Ik rekende er eens aan met behulp van deze te downloaden loop calculator:
http://www.dl0hst.de/magnetlooprechner.htm
Ik gebruikte 100W voor het vermogen. Daaruit blijkt dat de spanning over de loop ongeveer 6kV wordt. Dat is dus wat minder dan de 10kV waar Harry van uitging. De berekende capaciteit voor resonantie is volgens het programma 9,6pF en dat is ongeveer wat Harry aangaf en valt ruimschoots binnen het bereik van de pannensponscondensator. De efficiency van de loop is 1,6% (-18.1dB) en dat valt me eigenlijk een beetje tegen: mijn single turn loop met RG213 kabel van ongeveer dezelfde doorsnede heeft volgens hetzelfde programma 40% efficiency (dat is -4dB, dus iets meer dan een half S-punt). Met deze antenne verlies je 3 S-punten ten opzichte van een full size dipool. Dat verlies wordt voornamelijk veroorzaakt door het dunne draad dat gebruikt wordt: het scheelt nogal of je soepel installatiedraad gebruikt van 1,5mm diameter, of een RG213 kabel van 15mm diameter. Maar deze loop antenne is nog altijd beter dan geen antenne, en Harry claimt interessante resultaten te halen met slechts 5W. Soms is de praktijk beter dan de theorie. En aangezien de kosten van de antenne nihil zijn, is het de moeite waard het een keer te proberen.

Verhalen uit de werkplaats

In het verslag van onze Liechtenstein expeditie zijn twee voorvallen niet genoemd die misschien toch wel de moeite waard zijn om alsnog te vermelden. Degenen die ons volgen weten dat we meestal in Feldkirch dineren voordat we de laatste paar kilometers naar de hut afleggen. Daarna is het zaak om de auto's leeg te maken en zo snel mogelijk in de lucht te komen. Mans PA2HGJ en ondergetekende PA3CNO stortten zich op het ophangen van de Inverted-V in de lichtmast voor de hut. Hengeltje erbij, trekdraad over de bovenste sport van de lichtmast en het voedingspunt van de Inverted-V ophijsen. Snel naar binnen en verbindingen maken. Maar de signalen waren zwak, wat we in eerste instantie weten aan de condities. Pas later, toen we buiten gingen kijken, bleek dat we wel de paaltjes hadden neergezet om de uiteinden van de antenne aan vast te maken, maar dat in de haast niet hadden gedaan. De twee draden van de Inverted-V hingen gewoon langs de lichtmast naar beneden. Dat we desondanks toch nog verbindingen hebben gemaakt, is wederom een bewijs dat het nog heel moeilijk is om een antenne te maken die niet straalt...

Een tweede incident met de antenne deed zich wat later in de week voor. Ik had de antenne getuned op 80m en zat een reeks CQ's te geven in CW. Op een gegeven moment zie ik de SWR ineens fors oplopen tot boven wat de set nog lekker vindt. Ik dacht in eerste instantie dat er iets losgeschoten was, maar even later was de SWR weer goed. Maar na weer een riedel CQ's ging het wederom mis: de SWR schoot omhoog. Na overleg met de backup in Nederland (PA3HK) was de conclusie dat één van de ringkernen (het waren meerdere kernen van verschillende materialen) van het toegepaste

mantelstroomfilter niet tegen de frequentie van de 80m band kon. En inderdaad werden de ringkernen bloedheet, waardoor de magnetische eigenschappen veranderen en de SWR opliep. Vandaar de oplopende SWR. En toch bleef het voorval knagen. In principe wordt er immers geen veld opgewekt als de heengaande stroom gelijk is aan de teruggaande stroom. Daar kan - door onbalans in de antenne - wel iets verschil in zitten, maar niet zodanig dat een ringkern bloedheed wordt. Uiteindelijk haalde Mans PA2HGJ het voedingspunt van de Inverted-V naar beneden en wat bleek: één van de antennedraden was uit de kroonsteen geschoten. We waren dus op een halve antenne aan het zenden geweest, en ja, dan is de heengaande stroom significant anders dan de teruggaande stroom. Vandaar het hete mantel-stroomfilter... Nadat de antennedraad weer vastgezet was, werden de ringkernen niet meer heet. Zo zie je maar, je ziet de symptomen (hete ringkernen) maar de voor de hand liggende conclusie (antenne stuk) mis je compleet...

Hadden we het vorige keer al niet over storing? Ik zat te luisteren op 15m en die is bij mij best wel stil, in tegenstelling tot de lagere banden waar altijd wel een bult ruis staat. De voetbal-stations waren daar maar een S3-4, maar als ik ze op die banden kan horen, kan ik ze ook werken. Op een gegeven moment komt er een S5 aan storing te staan. Weg stations... Waar komt zo'n bak ruis ineens vandaan? Ik liep een rondje door het huis om te zien of er bij mij iets ingeschakeld was, voordat ik buren ging beschuldigen. Alleen de koelkast draaide. Het zal toch niet.... Ik schakelde de koelkast uit en checkte de storing op 15m. Stilte... Geen idee waarom een koelkast storing op HF moet veroorzaken, maar hij doet het wel. Vorig jaar om deze tijd heb ik de oude AEG vervangen omdat we een nieuwe keuken hebben laten plaatsen, en daar zit allemaal Whirlpool apparatuur in. Vanwege de matige condities had ik de hogere banden thuis nog niet veel gebruikt, maar dit is wel iets om uit te gaan zoeken. Misschien dat een simpel netfilter helpt...



Afdelingsnieuws

Over het afdelingsnieuws kunnen we kort zijn. De club is met zomerreces, en er gebeurt niet zoveel. Vanwege de vakanties is het avondrondje van 21:30 op de Zoetermeerse repeater verhuisd naar 7193kHz +/- QRM, waar de signalen wisselend zijn. Op het moment van dit schrijven zijn de condities om 21:30 doorgaans goed, om na een half uur in te storten. Dat hangt uiteraard af van het weer en de zonne-activiteit (of het gebrek daaraan).

Er wordt nog wel proefgedraaid met de APRS tracker V2, die over een duidelijk betere GPS beschikt. Als de testen goed verlopen, kan deze ook productierijp gemaakt worden en lees je het verhaal wel in de RAZzies. Robert PA2RDK heeft inmiddels ook APRS via LoRa draaien, wat misschien wel de toekomst heeft. Tenminste, als de dekking van LoRa wat beter wordt. Maar die is met APRS ook niet best: als je de rivieren oversteekt richting het zuiden, zit daar geen enkel APRS ontvangststation meer: de kaart is daar helemaal leeg. Maar daar hebben we nog wel ideeën over, hoe we dat kunnen verbeteren.

Ook in de maand augustus zijn er dus geen bijeenkomsten van de Radio Amateurs Zoetermeer. Pas op woensdag 12 september zijn we weer present. Tot die tijd: goede vakantie, en tot horens op de banden.